ELSEVIER

Robotics and Autonomous Systems 16 (1995) 213-245

Robotics and Autonomous Systems

Dynamics of behavior: theory and applications for autonomous robot architectures

G. Schöner [a,1], M. Dose [b,2], C. Engels [b,3]

[a] *C.N.R.S.-L.N.C., 13402 Marseille Cedex 20, France*

[b] *Inst. Neuroinformatik, Ruhr-Universität, 44780 Bochum, Germany*

Abstract

Limitations both for the further development as well as for the actual technical application of autonomous robots arise from the lack of a unifying theoretical language. We propose three concepts for such a language: (1) Behaviors are represented by variables, specific constant values of which correspond to task demands; (2) Behaviors are generated as attractors of dynamical systems; (3) Neural field dynamics lift these dynamic principles to the representation of information. We show how these concepts can be used to design autonomous robots. Because behaviors are generated from attractor states of dynamical systems, design of a robot architecture addresses control-theoretic stability. Moreover, flexibility of the robot arises from bifurcations in the behavioral dynamics. Therefore techniques from the qualitative theory of dynamical systems can be used to design and tune autonomous robot architectures. We demonstrate these ideas in two implementations. In one case, visual sensory information is integrated to achieve target acquisition and obstacle avoidance in an autonomous vehicle minimizing the known problem of spurious states. In a second implementation of the same behavior, a neural dynamic field endows the system with a form of obstacle memory. A critical discussion of the approach highlights strengths and weaknesses and compares to other efforts in this direction.

1. Introduction

Although there are probably at least as many definitions of the concept "autonomous agent" as contributions to this special issue, two fundamental requirements are, we believe, generally agreed upon: (1) Autonomous agents structure their behavior on the basis of sensory information that they themselves acquire. (2) Such autonomy goes beyond the sensor-driven nature of control systems in that it is flexible. Minimally this means that an agent may change its behavior qualitatively under the influence of sensory information. Flexibility in this sense comprises switching among different elementary behaviors, adapting behaviors in structured environments, learning new behaviors, and maintaining behaviors in the face of structural change of the system itself (robustness).

There is a latent, but inherent, conflict between these two requirements: Basing one's action on sensory information implies that one establishes a continuous link between sensing and acting. Flexibility implies that such a link must be relinquished at times in order to free the system to establish new and different relationships between sensing and acting.

If one looks at the history of autonomous robotics from this angle, one might interpret the two main lines of research, the control theoretic and the artificial in-

[1] E-mail: gregor@lnf.cnrs-mrs.fr

[2] E-mail: dosi@neuroinformatik.ruhr-uni-bochum.de

[3] E-mail: christo@neuroinformatik.ruhr-uni-bochum.de

SSDI 0921-8890(95)00049-6

telligence approaches, to flow from two different attitudes taken with respect to this inherent conflict. In control theory, goals are achieved by establishing a particular link between system parameters and sensed consequences of the system's action. Thus, no need arises to extract a priori the relevant information about a particular goal from the sensory array. For example, a vehicle can be guided to a particular location by reducing the sensed deviation from this location to zero, without ever determining from the sensor readings the actual position in space of the vehicle (e.g., [38]). Within this approach it is quite difficult, although not impossible, to simultaneously achieve several different goals, to flexibly change goal, or to generate behavior that is not so directly linked to on-line sensory information (for example, navigating based on memorized information that is not currently available at the sensory surface).

By contrast, the cognitive or artificial intelligence approach has addressed from the outset the need to separate the system from its environment by introducing representations of the world, which are invariant under important classes of transformations and actions. For instance, a symbolic description of a work space in terms of extereospecific object descriptions is invariant under such operations as moving the observer around, changing lighting conditions (for optical sensors), or any other change in those dimensions of the sensing-acting cycle that are not included in the symbolic description. Action is then planned at the level of such invariant representations and plan execution is essentially open loop. Flexibility is automatic in such a scheme, in that one may take many different attitudes to represented information simply through algorithms that include decision points. For example, an object may either serve as target, as tool, or as obstacle, and corresponding action plans may be derived. On the other hand, the notorious difficulty to generate and update adequate representations of the world on the basis of sensor readings makes it much more difficult to maintain a link between acting and sensing in a natural (that is irregular and changing) environment.

Still from this particular viewpoint, the behavior-based approach [9,12] could then (only in hindsight, of course) be interpreted as an attempt to return to control theoretic ideas. The individual elementary behaviors that robots are to be built up from, can essentially be viewed as control systems. They do not require extraction of invariant information and representations, because they are not flexible by themselves. Instead, special purpose control-like linkages between sensors and actuators can generate basic behaviors. The new idea in the behavior-based approach is to generate flexibility by having behaviors interact with each other. In the more rigorous formulation, this interaction is to be free of exchange of information (so as to strictly avoid the extraction of invariants) and limits itself to turning on and off behaviors (which includes, however, the ability to use or subsume behaviors).

The interaction among elementary behaviors remains the Achilles heel of behavior-based systems. This problem is sometimes called the problem of architecture and a number of proposals has been made to overcome it [11,5,3,40,54]. These range from radical ideas that minimize the notion of state and operate in pure feed-forward fashion [14] to ideas compatible with cognitive approaches such as superposition of vector fields generated by schemata [4]. We believe that it is at this point that lies the fundamental limitation of our current methods. The absence of clear theoretical concepts for this architectural problem hinders the further development of autonomous agents in two ways: First, by making it difficult to scale up the number of elementary behaviors and the number of corresponding sensor and effector subsystems. Second, by making it difficult to attain more invariant (i.e., more cognitive) types of behaviors such as the capabilities to memorize, to fuse sensory information from different sources, or to generalize.

The aim of this paper is to discuss a particular theoretical language in terms of which one might attempt to build such architectures. There are three elements of this language: (1) the concept of behavioral variable; (2) the concept of behavioral dynamics from which behaviors are generated through attractors; (3) the principle of neural representation of information, which can be made dynamic by allowing for the self-generation of neural activation patterns. In Section 2 we introduce the theoretical ideas in a tutorial form. Two exemplary applications of these ideas are implemented on a mobile platform equipped with a vision system (Section 3). Extensions, the relation to other work, strengths and drawbacks are covered in Section 4.

2. Dynamics of behavior

The theory of dynamical systems as such is, of course, a field of mathematics. A scientific theory making use of that field must map problems from the specific domain of natural or engineering science onto the mathematical concepts. In this section we sketch a particular such mapping.

We do this quite simply by lifting ideas from a particular approach to behavior of nervous systems, which stresses the concepts of pattern and temporal order (review [50,47]). This work stands in the tradition of biological cybernetics (see [37] for review; [46] for a comparison) and ecological psychology [60], and has ties to developmental psychology [58], neurobiology [16] and perception [25]. Specifically, three ideas are borrowed from there: (1) It is useful to measure the activity of behaving systems through variables which adequately express those aspects of behavior that are invariant under certain changes of environmental conditions (pattern variables). (2) Behavior can be conceived of as resulting from a dynamical system that the nervous system is able to establish. This means, that the nervous system not only generates particular individual time courses of pattern variables, but also the neighborhood of those time courses. For instance, the recovery of a behavior following perturbation can be conceived of as resulting from the asymptotic stability of the attractor solutions of an underlying dynamical system. (3) Sensory (or internal) information affects the dynamics, not directly the patterns [51]. In addition, we introduce the concept of neural representation with topology and strong cooperativity [49,32], which is related to much current work in neural network theory [1,45]. The goal of this section is to explain the concepts alluded to here and to show how they can be used constructively to design behaving systems.

(1) Behavioral variables. The first step towards designing a behavior is to find variables (possibly vector-valued) to describe it. In a sense, the variables define behavioral dimensions, that is, continua along which the behavior can change. A specific instance of the behavior then corresponds to a point in this space of behavioral dimensions. In other words, all variables have a particular value for any specific instance of behavior. We shall call these sorts of variables *behavioral variables.*

Two examples will be used throughout this section to illustrate the ideas. First, consider movement of an autonomous vehicle in the plane. Such movement will be controlled, so that, for instance, particular target locations are reached and locations of obstacles are avoided. To describe the behavior that this movement represents, we might use the heading direction, ϕ, of the vehicle and its velocity, v, as behavioral variables. These variables represent the dimensions along which the movement behavior can vary. At each point in time, an autonomous robot must provide particular values for these variables.

As a second example consider a somewhat more abstract behavior: the same vehicle is to represent its position in the world based on various types of sensory information. Remember that we are attempting here to treat all processes occurring along the stream from sensing to acting as behaviors, even when this means stretching the common usage of the term "behavior" a bit. Along which dimensions can the behavior of estimating one's position in the plane vary? Quite simply, along the spatial dimensions of the plane. An adequate variable is, therefore the vector, $\boldsymbol{r}_{\text{ego}} = (x_{\text{ego}}, y_{\text{ego}})$ describing the current estimated ego-position in a world coordinate system.

Not just any set of variables will do. A specific requirement arises from the need to endow the particular behavior with tasks, goals, or objectives. It must be possible to express such tasks as particular values or sets of values of the variables. More abstractly, tasks must be expressible as points or sets in the space of behavioral dimensions spanned by the behavioral variables. This means, in particular, that the points representing the tasks must not depend on the current state of the system, that is, must be independent of the values the behavioral variables have at the moment.

In the first example objectives are, let us say, to move toward a target while avoiding to run into obstacles. In terms of the behavioral variable heading direction, ϕ, these tasks can be expressed as particular values: The heading direction, ψ_{tar}, points toward the target location from the current vehicle position, and the heading directions, ψ_{obs}, point toward the obstacle locations from the current vehicle position (top part of Fig. 1). Note that directions, ϕ, ψ_{tar}, and ψ_{obs} are relative to an allocentric reference direction (in the top part of Fig. 1 chosen as the direction parallel to the x-axis of a world coordinate system). Therefore, the

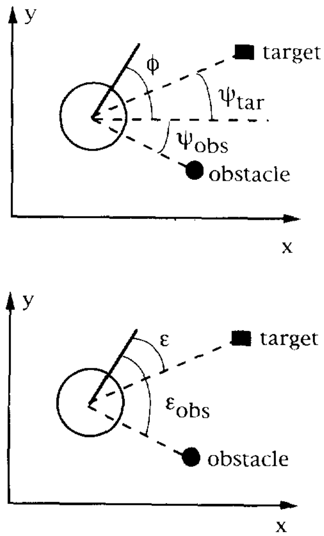


Fig. 1. The task of moving in the (x, y) plane toward a target while avoiding obstacles: The choice of behavioral variable illustrated on top is adequate: Vehicle motion is controlled by heading direction, ϕ, relative to a world coordinate axis, here the x-axis. The tasks are then parametrized as particular values, ψ_{tar} and ψ_{obs} of heading direction which are independent of the current heading of the vehicle. The choice of behavioral variable illustrated on bottom is compatible with control-theoretic approaches, but not with the dynamic approach. Here, the deviation, ϵ, of vehicle heading from the direction toward the target might be directly available from typical sensor systems. However, other tasks such as avoiding obstacles at directions, ϵ_{obs} are not independent of the current state of the vehicle: As ϵ changes so does ϵ_{obs}.

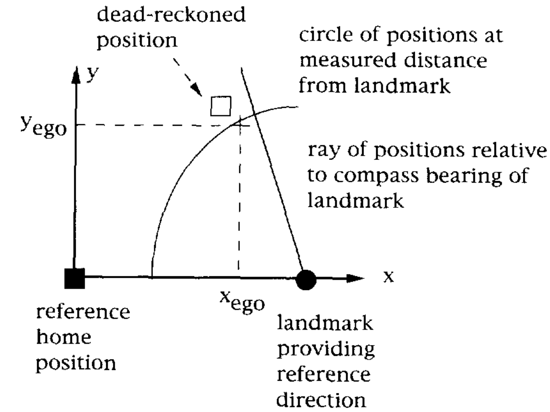


Fig. 2. The task of estimating ego-position, (x_{ego}, y_{ego}), relative to a reference coordinate system (here defined by a home base and a reference landmark). Three sources of information on ego-position are indicated: A circle as obtained by measuring distance from the landmark, a ray obtained by measuring the bearing of the landmark relative to a compass (which is calibrated in the reference coordinate system) and a small square representing the dead-reckoned position with finite uncertainty. All sources of information can thus be transformed into sets of points in the space of the behavioral variables (x_{ego}, y_{ego}).

directions toward the target, ψ_{tar}, and toward the obstacles, ψ_{obs}, are independent of the current heading, ϕ, of the vehicle. If the behavioral variable had been chosen as, for instance, the deviation, $\epsilon = \phi - \psi_{tar}$, from the heading direction toward the target (an obvious choice from the point of view of control theory), this would not be true: Now the heading direction, ϵ_{obs}, toward an obstacle depends on the variable, ϵ! Therefore, this is not a consistent choice of behavioral variable.

Consider also the second example from this angle (Fig. 2): The objective is quite trivially to obtain an adequate estimate of ego-position, clearly a point in the space, (x_{ego}, y_{ego}). More precisely, the definition of this space is relative to some allocentric references (e.g., a home base and a land mark) and this determines in which sense ego-position must be estimated (namely, relative to that reference frame). Sensory sources of information may come in different formats originally. For instance, a landmark detector may provide a distance from the landmark which defines a circle in the space of ego-position. Jointly with a compass a direction relative to such a landmark may be obtainable leading to a ray in the space of ego-position. Dead-reckoning might provide a complete estimate, a point in ego-position space. When these sources of information as expressed as sets of points in the space of the behavioral variable, they are brought into a format in which they can be integrated.

(2) Behavioral dynamics. The next step is to set up a dynamical system, the solutions of which generate behavior in time. The dynamical system is simply an equation of motion of the behavioral variables. For instance, for heading direction a dynamical system defines the rate of change of heading direction, $\dot{\phi}$, as a function of current heading direction, ϕ

$$\dot{\phi} = f(\phi) \tag{1}$$

and solutions, $\phi(t)$, of this equation represent the ongoing behavior of moving around in the plane. (To simplify things we shall assume from now on that the other behavioral variable, velocity, has some fixed, non-zero value.) In the second example the equation of motion is vector-valued:

$$\dot{x}_{ego} = f_x(x_{ego}, y_{ego}) \quad (2)$$

$$\dot{y}_{ego} = f_y(x_{ego}, y_{ego}) \quad (3)$$

We use the term *behavioral dynamics* to refer to these kinds of differential equations. This form of dynamics is totally unrelated to the mechanical dynamics of robot systems [4].

Not just any dynamical system will do, nor are we interested in just any kind of solution. Two more specific requirements are: (1) The dynamical system must be dissipative and have asymptotically stable fixed points or other limit sets. (2) Behavior must be generated through attractor solutions, so that the system is at all times in an attractor state. To explain these requirements we first give a few definitions, and then go through the two examples.

Our use of dynamical systems takes reference primarily to the qualitative theory of dynamical systems. Numerous excellent textbooks introduce into this area. We mention [41] at an advanced level and [10] at an elementary level (for an informal tutorial appealing to intuition see also [29]). Here we provide only the most minimal background.

The space of dynamic variables is called *phase space* or state space. In the first example, the phase space is made up of a single dimension, ϕ, but in the second example, it has two dimensions, x_{ego} and y_{ego}. Dynamical systems such as Eqs. (1) and (2,3) define a vector at each point in phase space (in the examples, $f(\phi)$ and $(f_x(x_{ego}, y_{ego}), f_y(x_{ego}, y_{ego}))$, respectively). These vectors determine the direction and rate in which the system will move from each point in phase space. The ensemble of these vectors is called the *vector field*. We are interested in a particular type of solution of dynamical systems, called *fixed points*, at which the rate of change of the variable is zero. These are the zeros of the vector field, e.g., $\dot{\phi}_{\text{fixed point}} = 0 = f(\phi_{fixed\ point})$. Such fixed points are, in other words, constant solutions of the dynamical system. A fixed point is *asymptotically stable*, if the system converges in time to the fixed point from points nearby. An asymptotically stable fixed point

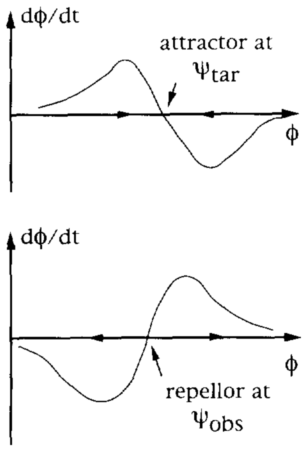


Fig. 3. Basic concepts of dissipative dynamics: The rate of change, $\dot{\phi} = d\phi/dt$ is plotted as a function of ϕ. The points at which $d\phi/dt$ is zero are fixed points of the dynamics. If the rate of change has negative slope at the fixed point (top) then this fixed point is an attractor, to which solutions starting nearby relax as indicated by the arrows. Reversely, if the slope of the rate of change is positive at the fixed point the fixed point is a repellor.

is an example of an *attractor*, a term we use mostly for its intuitive appeal: the asymptotically stable fixed point "attracts" solutions in its neighborhood in the course of time. Repellors are fixed points who attract if time runs backwards. In other words, solutions diverge from points in the neighborhood of repellors (except if started exactly on the fixed point).

These concepts can be generalized to more complicated sets, for example, to periodic solutions, but we shall not go beyond this simplest case in this paper. Because all points in an entire neighborhood of a fixed point attractor converge to the fixed point, the volume of this neighborhood (or area or length, depending on dimension) is reduced over time to zero. This shrinkage of volume of sets of initial points as the solutions evolve defines the dynamical system as a *dissipative* dynamical system. Control-theoretic stability is mathematically linked to these concepts of asymptotic stability and dissipation (and we will sometimes omit the qualifier "asymptotic" of stability even though this is, strictly speaking, incorrect usage).

In Fig. 3 these concepts are illustrated in terms of the first example. The dynamics of heading direction are shown by plotting the rate of change of heading direction, $\dot{\phi} = d\phi/dt$ as a function of heading direction, ϕ. Intersections of this function with the ϕ-axis are

[4] Indeed, we shall clarify further on that the behavioral dynamics must always be dissipative, while the mechanical dynamics are typically considered in the frictionless limit in which they define conservative (or Hamiltonian) dynamical systems.

fixed points. A negative slope of the function $d\phi/dt$ at the fixed point characterizes an asymptotically stable fixed point or attractor (top part of the figure). To see this, consider values slightly to the right of the fixed point. At these points a negative rate of growth drives the system back to the fixed point. Analogously, solutions starting to the left of the fixed point increase toward the fixed point. Thus, to make the direction ψ_{tar} an attractor, the designer needs to erect a vector field with a zero at ψ_{tar} and negative slope there (top of the figure). The slope of the function, $\dot{\phi}$, determines the rate with which the system relaxes to the attractor. The steeper this slope, the stronger the restoring forces and the faster the system relaxes to the attractor. Because relaxation is exponential in time, it can be characterized by a time scale, the relaxation time, τ_{rel}, defined as the reciprocal inverse of the slope of the dynamics at the fixed point:

$$\tau_{rel} = -\left[\left. \frac{df(\phi)}{d\phi} \right|_{\phi=\phi_{fixed\ point}} \right]^{-1} \qquad (4)$$

If put at a distance, ϵ, from the attractor, the system reduces this distance by a factor of $e = 2.7...$ within one relaxation time.

Reversely, a repellor can be constructed at a direction, ψ_{obs}, pointing to an obstacle by erecting a vector field with a zero there but positive slope (bottom part of the figure).

Consider now the dynamics with a single attractor at ψ_{tar} (top diagram in Fig. 3). From initial values in the neighborhood of the attractor the system has relaxed to the attractor after a few relaxation times. This heading direction, ψ_{tar}, now determines the robot's movements. As the robot advances (at constant velocity), it changes its position relative to the target (top of Fig. 4). As a result, the direction pointing from the vehicle toward the target changes: $\psi_{tar} = \psi_{tar}(t)$. This change gradually shifts the attractor in the space of heading direction (bottom diagram of Fig. 4). However, if the change occurs sufficiently slowly (compared to the inner time scale τ_{rel}) then the system can relax to the new attractor position immediately as this shift occurs. In this case, the system essentially always sits in the attractor and moves with the attractor as the vehicle moves. This is the limit case that we shall be using to design behavioral dynamics. It can always be achieved by adequate choice of the time scale, τ_{rel}, of

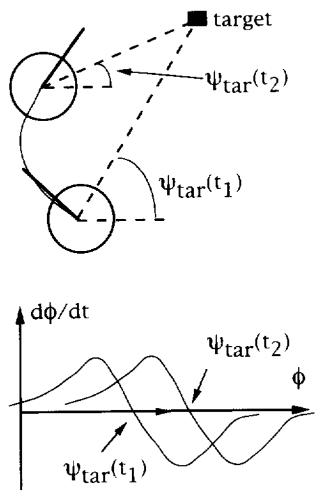


Fig. 4. As the vehicle moves in the plane, the direction, ψ_{tar}, in which the target lies changes (top) This change shifts the attractor of the heading direction dynamics (bottom). If this shift occurs more slowly than the relaxation process of that dynamics, then the system is at all times in the attractor of the heading direction dynamics.

the behavioral dynamics (or by driving slowly)[5].

The second example cannot be illustrated so simply, because now a two dimensional vector field must be erected. The concepts sketched here generalize to multiple dimensions. Stability can be decided by linearizing the vector field around the fixed point and determining the eigenvalues of the Jacobian matrix. The real parts of these eigenvalues play the role of the slope above: they must all be negative for asymptotic stability. The least negative real part represents the direction along which the system relaxes most slowly. This component dominates the overall relaxation process, and relaxation time is thus defined as the inverse reciprocal of that least negative real part of an eigenvalue.

So far we have only looked at a single task at a time, defining individual attractors or repellors to im-

[5] In this limit case the heading direction in the figure should have immediately relaxed to ψ_{tar} and the robot should have moved straight toward the target rather than on a curved path. On that straight path, ψ_{tar}, would not actually change. To illustrate the principle we show a curved path, on which this angle changes. In general, the presence of other contributions to the behavioral dynamics (such as obstacle avoidance) will lead to curved paths, because the attractor does then not always coincide exactly with ψ_{tar}.

plement such a task. This might still be considered a mere rewrite of control theory. To go beyond control we must address how multiple tasks and multiple sources of sensory or internal information can be integrated. At the same time, this will pose more seriously the problem of how to arrive at mathematical functional forms for the behavioral dynamics and how these dynamics are linked to incoming sensory information.

The dynamics is build up from individual *contributions*, which are added to form the total vector field [6]. Each contribution represents a constraint on the behavior that we are designing. These constraints typically arise from sensory information, but may also be built-in as fixed goals, for instance, or fed in from internal memory representations. Each contribution is characterized in three respects (Fig. 5): (1) Which behavior is specified by the contribution? (2) How strong is the contribution? (3) Over which range of the behavioral variables does the contribution exert influence? We explain these points in detail next.

By choice of the behavioral variables, tasks can be expressed as points or sets of points in the phase space of the behavioral dynamics. We now assume that sensory channels or internal representations provide information about these points. They may completely specify a particular point or only provide a bound, that is, an area in phase space. This is, in a sense, the "contents" of sensory or internal information. A contribution to the behavioral dynamics is now designed such, that *if it were the only contribution to the behavioral dynamics* it would erect an attractor at the specified point (for information about to-be-achieved behavioral states) or erect a repellor there (for information about to-be-avoided behavioral states). If sensory or internal information provides only a bound on the desired regions of the behavioral phase space, then the contribution is defined such that again in isolation from other contributions it would attract toward the region in which the to-be-achieved behavior lies and repel from the region in which it does not lie (and reversely for information about to-be-avoided behaviors). This can be achieved in various ways: An individual representative point within the specified region may be selected and an attractor or repellor be erected there. The range of attraction/repulsion must be chosen to adequately reflect the bound provided by the sensor (see below). Another possibility, more sophisticated, is to erect a set of fixed points in the entire region specified. These can be made attractive relative to the neighborhood of the region, but marginally stable within the set of fixed points. (Marginal stability means that there are no restoring forces to perturbations that put the system into another fixed point.). Fig. 5 illustrates a contribution that erects a single attractor (top) and one that makes an entire region attractive (bottom). In either case, sensor readings are continuously fed into the behavioral dynamics as those parameters, that fix where the attractors or repellors come to lie.

How strongly the specified behavioral state is stabilized by a contribution is best characterized through the time scale, τ_{rel}, generated by the contribution alone

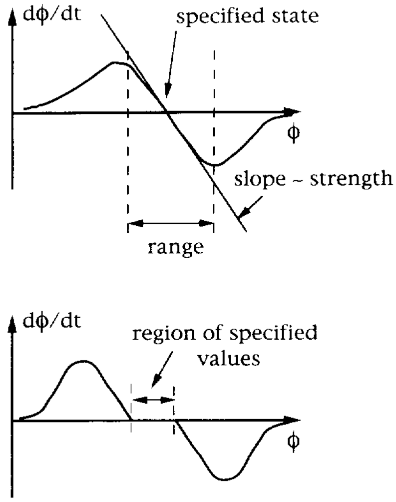


Fig. 5. Concept of a contribution to the behavioral dynamics. Top: Each contribution is characterized by (a) the state it specifies as an attractor (shown here) or repellor, (b) the strength of attraction or repulsion parametrized by the slope of the dynamics at the fixed point, and (c) by the range over which attraction or repulsion is effective. Bottom: If a set of values is specified, the contribution erects attractive (shown here) or repulsive forces outside the set. The vector field is zero within the set. An alternative is to proceed as on top, but with a "default" value selected from within the set as the specified state. In this case the range must be chosen to reflect the extension of the specified set.

[6] At this point additivity at the level of the vector field is no limitation whatsoever, because there are no constraints on the functional forms of contributions. Thus, for instance, non-additive effects can be brought about simply by writing down the non-additive function as a separate contribution. Here, additivity is a matter of defining the concept of contribution.

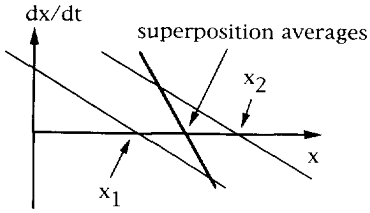


Fig. 6. When linear contributions to a behavioral dynamics (thin lines) are superposed (bold line) the attractor lies at a position that is the average of the attractors specified by the individual contributions.

(that is, in the absence of other contributions). For an attractor this is the relaxation time discussed earlier. For a repellor, an analogous time scale can be defined as a reciprocal escape rate. The shorter these times, the more strongly attractive or repulsive the specified state. For a single variable, this strength of attraction or repulsion can be visualized by the slope of the dynamics at the fixed point (top of Fig. 5): The steeper the vector field at the fixed point, the stronger the corresponding contribution. Other than contributing to the overall time scale of the behavioral dynamics, the concept of strength of a contribution is crucial to determine how multiple contributions interact. We return to this below.

The need to address the range over which contributions extend arises primarily as a question of how various contributions interact. For a single contribution an infinite range does not cause any particular problem. If more than one contribution are present, however, infinite range always leads to interaction. We use our second example to illustrate the problem (Fig. 6): Consider two contributions erecting attractors at two estimates, x_1 and x_2, respectively, of ego-position x (we use only the x-coordinate for this illustration). In the figure the contributions are taken as linear functions, intersecting at the specified points, x_1 and x_2, with negative and equal slope. The linear functional form implies an infinite range. Superposition of these two contributions gives rise to a (steeper) linear function, which now intersects at the arithmetic mean of x_1 and x_2. (Had we chosen different strengths, then the attractor of the complete dynamics would lie at a correspondingly weighted mean.) Thus, the two contributions always "interact" by averaging irrespective of the distance between the two estimates, x_1 and x_2.

Endowing the contributions with a finite range (see top of Fig. 7) allows us to modulate the way the two contributions interact (middle and bottom part of Fig. 7). At small distances between the estimates, the superposed contributions lead to a single, averaged attractor (middle part of the figure), while at large distances, the contributions lead to two separate attractors at the two specified locations. In other words, the range of contributions in phase space determines the amount of overlap between pieces of information that will lead to averaging or fusion of the information. More generally, task demands are implemented independently when they overlap less than the ranges of the corresponding contributions, and are implemented as averaged task demands when they overlap more than the ranges of the corresponding contributions. Note that limiting the range of contributions necessarily leads to non-linear dynamics!

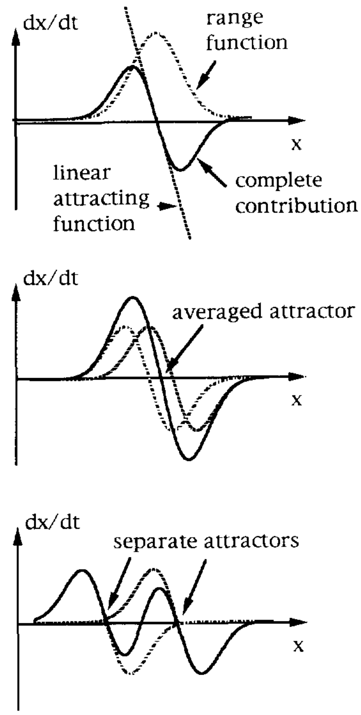


Fig. 7. The dynamics of ego-position estimation (x-component only). When for each contribution a linear dynamics is multiplied with a range-limiting function (top), the superposed dynamics can either perform averaging (middle: the two contributions are dashed and dash-doted, the superposition is the solid line) or maintain separate attractors (bottom) depending on the distance between the specified states of the individual contributions.

Based on these three criteria, the functional form of contributions to a behavioral dynamics can be constructed as follows: (1) Determine the point or region in phase space that is specified by the particular source of information. (2) Write down a linear function whose zero is located at the specified point in phase space. For a region, linear functions depart from the boundaries of the region. In more than one dimension, a linear function can be applied in each direction independently (diagonal form of the dynamics) or polar coordinates can be used. (3) The slope of the linear functions is chosen negative for to-be-achieved states and positive for to-be-avoided states. The absolute value of the slope is chosen based on considerations of time scale and of relative dominance of contributions, the more dominant contributions having larger absolute value. This factor can be tuned to represent confidence in a particular source of sensory information. (4) These linear functions are multiplied with a range-limiting function. A gaussian profile centered on the specified point with amplitude equals 1 is best suited, because it does not change the slope of the resultant dynamics from that of the linear function. In actual applications (see Section 3) it is often better to cut the long tail of such gaussian range functions by a sharper threshold function in order to avoid the interference of these tails with other contributions. (5) The contributions are added to obtain the behavioral dynamics.

Although the logic behind these steps is general, the concrete functional form should be adjusted to the needs of the problem at hand. For instance, in the first example we may design the dynamic contributions for obstacle avoidance using sine-functions instead of linear functions so as to automatically fulfill the periodicity requirements of the dynamics of heading direction (that is, the requirement that the dynamics is the same again after the robot has made a full 360-degrees turn). Fig. 8 illustrates how this contribution can be constructed, for instance, as $f_{\text{obs}} = \sin(\phi - \psi_{\text{obs}}) \exp[-(1 - \cos(\phi - \psi_{\text{obs}}))/2\sigma^2]$ (in Section 3 we use a slightly more sophisticated version, however).

Fig. 9 shows how the interplay between two such contributions leads to flexible behavior. In a sense, the system becomes able to make a decision of whether to drive through between two obstacles or whether to go around. What is varied from top to bottom, is the angular distance between the directions in which the two

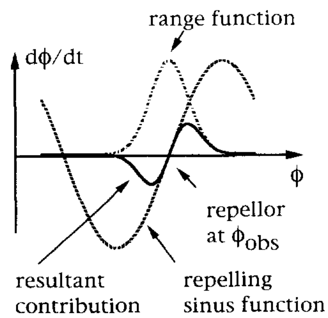


Fig. 8. The obstacle contribution to the heading direction dynamics: A – sin-function (dashed) intersecting at ψ_{obs} with positive slope is multiplied with a range-limiting function (dash-dotted) to obtain the complete contribution (solid) of finite range.

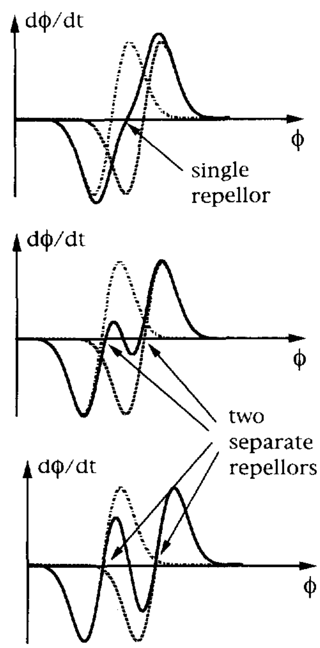


Fig. 9. Decision making in the heading direction dynamics: From top to bottom the distance between two directions in which obstacles are detected increases. The two individual contributions (dashed and dashed-dotted lines) overlap to generate (solid line) a joint repellor at small distances (top), but then near a bifurcation (middle and bottom) the repellor splits into an attractor and two separate repellors. The case on top corresponds to a decision of the vehicle to circumnavigate the two obstacles, while in the other examples passing in-between the two obstacles is a possible (but not the only) solution.

obstacles are seen. When that distance is sufficiently small (top part of figure), the contributions overlap strongly and a single repellor is erected at an averaged direction. The range of repulsion is adequately broad. At a critical distance between the repellors, a bifurcation or instability occurs (middle part of figure). The repellor splits into two repellors and an attractor in between. At this point, the system can begin to drive through, although the range of initial heading directions from which this choice will be made is small (attractor has a small basin of attraction). At larger separation, the solution that allows passage between the obstacles becomes quite attractive (bottom part of figure).

Another way to visualize what happens here is to look at a *bifurcation diagram* (Fig. 10, top). The positions of attractors and repellors are plotted as a function of an external parameter that brings about a decision. Here, the angular distance, $\Delta\psi_{\text{obs}}$, between the directions in which two obstacles are sensed is such a parameter. At small distances, a single repellor prevails from the overlapping obstacle contributions. At a critical value of $\Delta\psi_{\text{obs}}$ the repellor splits into two repellors and an attractor. This is a so-called subcritical pitchfork bifurcation (the name alluding to the graphical appearance of the bifurcation diagram). At such a bifurcation, the repellors and attractors that collide go through an instability, that is, one of the eigenvalues of the linearized dynamics has zero real part, corresponding to infinite relaxation time.

Instabilities can be analyzed analytically or, more often, numerically. The fixed points and their eigenvalues can be determined as parameters are varied. The zeros of an eigenvalue (of its real part to be precise) are the bifurcations points. Parameters of the dynamics determining the range of contributions can then be adjusted to make a bifurcation appear at an adequate point. For instance, in this example the designer may want to adjust the range functions such that the pitchfork bifurcation occurs at that angular separation between two obstacles, at which the vehicle will just fit physically in-between the two obstacles (this value will have to be distance dependent). This is how we adjusted parameters in our implementations (Section 3). In practice, a complete analysis is not always necessary. It is often sufficient to obtain information about the layout of attractors and repellors at just a few settings of sensory input.

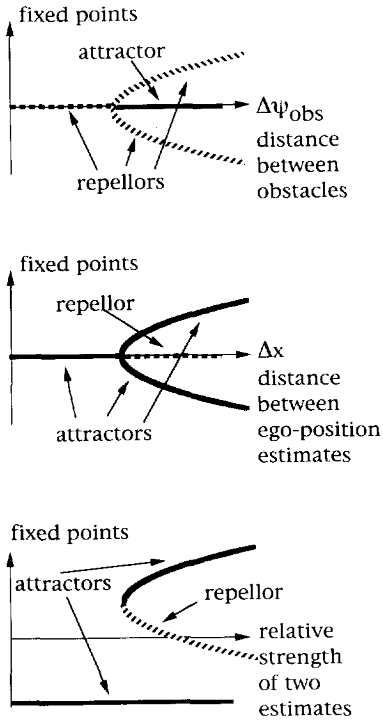


Fig. 10. Bifurcation diagrams: Fixed points are plotted as functions of external parameters. Solid lines indicate attractors, dashed lines indicate repellors. Top: Subcritical pitchfork bifurcation as it occurs in Fig. 9 as a function of the angular distance, $\Delta\psi_{\text{obs}}$, between two detected obstacles. Middle: Supercritical pitchfork bifurcation as it occurs in the sensory fusion dynamics sketched in Fig. 7. Bottom: An alternative bifurcation for this dynamics is the tangent bifurcation occurring here in the upper part of the figure. An attractor and a repellor collide and annihilate as the relative strength of two contributions is varied. A remote attractor is unaffected. The corresponding dynamics are illustrated in Fig. 11.

One can use this form of bifurcation analysis more radically to design a system around particular bifurcations. For instance, in the second example, the decision to average two estimates of ego-position or to base the representation of ego-position on only one estimate, comes about by a very similar bifurcation (see Fig. 7 for the dynamics and the middle part of Fig. 10 for the bifurcation diagram). This is essentially the same as the pitchfork obtained for obstacle avoidance, but now with attractors and repellors interchanged (a so-called supercritical pitchfork bifurcation). This bifurcation leads to bistable behavior when the two sources of information are not fused. Which state is realized then depends on the previous history of the system. It

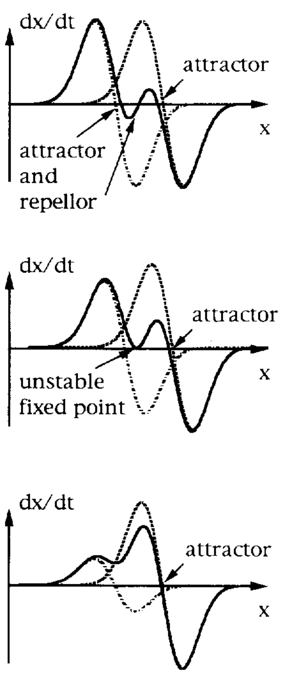


Fig. 11. Tangent bifurcation in the dynamics of ego-position estimation: From top to bottom the relative weight of two contributions (dashed and dash-dotted lines) is varied. When both contributions are equally strong (top) the system is bistable. As the left contribution loses strength, the corresponding attractor becomes unstable until it collides with a repellor and at that point is tangent to the ϕ-axis. The attractor corresponding to the left entry has disappeared after the strength of that contribution has been further reduced (bottom).

will tend to stick with a previous representation until the attractor corresponding to that information becomes unstable (so-called *hysteresis* behavior). This might be the desired behavior under circumstances where no rational choice between the two sources of information can be made. Under other circumstances, however, the designer may wish to eliminate one of the attractors entirely, for instance, because the quality of pertinent sensory information is lower for one than for the other attractor. This alternative can be realized through a *tangent bifurcation* illustrated on bottom of Fig. 10. In the tangent bifurcation an attractor collides with a repellor at a particular parameter value leaving no trace beyond the bifurcation point. Other attractors may continue to exist unaffectedly. In Fig. 11 we illustrate how a tangent bifurcation can be brought about in the dynamical system of an ego-position estimation system. In this case, two contributions specify attractors at a constant distance from each other. What is varied in the series from top to bottom is the relative strength of these two contributions. As the left-most attractor becomes less stable, the overlap with the dynamic contribution on the right pushes repellor and attractor to each other. The two collide in the middle frame and have vanished on bottom. An application of these ideas to sensory fusion for the representation of home position can be found in [39].

Other bifurcation types, such as the transcritical or the Hopf bifurcation can be similarly employed where adequate. In the style sketched above, bifurcation theory, which provides a classification of local bifurcations and the local functional forms that robustly generate these bifurcations, can be used to design the behavioral dynamics.

Other than these practical implications this linkage of bifurcations with decision making offers an operational definition of flexibility in autonomous systems. Flexibility occurs if gradual change of sensory information can lead to qualitative change of behavior. Qualitative change can now be defined quite literally in the sense of the qualitative theory of dynamical systems as change during which the number, nature or stability of attractors and repellors is changed [7].

(3) Neural field dynamics. There is a fundamental limitation of the concepts we have developed so far. Dynamical systems have a unique state at all times. They can change state by continuously moving in state space. This is quite adequate for the control of an effector system, which physically has these same two properties. But how about representations of information? Our example of representation of ego-position illustrates this. The dynamical systems formulation we gave forces a unique ego-position estimate at all times, which will evolve continuously in time (although change can be rather quick near a bifurcation). This may be fine for ego-position where one would want to force an estimate even in the absence of sufficient information. But how about using these ideas to estimate, for instance, the optic flow field or

[7] The more general formulation of qualitative change involves change of the ensemble of solutions of a dynamical system that is not topologically invariant. Thus the ensemble of solutions before a bifurcation cannot be continuously deformed into the ensemble of solutions following a bifurcation.

for representing the locations of obstructions in the path of the vehicle. There might be conditions under which one might want to proceed in a similar fashion, for instance, if an estimate of optic flow has already been obtained and is to be updated continuously in time. However, more generally the assumption about a unique state of the dynamical system makes no sense. For instance, the number of detectable obstructions may vary. Even if we assign a new variable to each detected obstruction, we could not possibly maintain temporal continuity because we would not be able to assign new incoming sensory information to the correct variable (without solving at the same time a big computational problem, the matching problem).

More generally, this problem arises as we try to address processes such as memory, planning, or the representation of sensory information. Can the concepts of dynamics be carried over into the domain of these processes? We ask this for two reasons: First, as we stated at the outset, a unified language for all levels of an autonomous system is an important prerequisite to providing integrative architectures for such systems. Second, some of the properties of dynamics may actually be very useful within such processes, essentially to provide these processes with well-defined time scales and stability properties, so that their behavior relative to time-varying information can be designed.

The new ingredient needed is the principle of neural representation. The idea is to introduce an additional "auxiliary" variable, which we shall call activation (the analogy being with neural excitation). The behavioral variables are now not by themselves dynamic, but instead serve as indices of a field of activation variables. In other words, with each value of the behavioral variable is associated an activation variable. The activation "represents" that value of the behavioral variable. Strong activation indicates the presence of the represented value (presence in the input or in the output, depending on what the function of the behavioral variable is). Weak activation indicates the absence of the represented value. This is the form in which information is represented in higher parts of the nervous system according to a common hypothesis. In neuroscience, this principle is referred to as space code (or, historically, as the principle of equivalent nervous energy). Thus, the term neural representation seems adequate, even though the analogy with neurophysiology is at a rather formal and abstract level.

Three ideas are needed to make the principle of neural representation compatible with the concepts of dynamics: (1) topology; (2) neural dynamics; (3) self-generation of activation.

A topological neural representation is one in which activation variables have well-defined neighborhood relationships. Here we consider topologies induced by the space that is represented, that is, induced by the behavioral variables. Because these variables are continuous in nature, they define a natural topology. Consider, for instance, the neural representation of heading direction, ϕ, in the first example. A neural activation, u, is defined for every value of heading direction, so that we obtain a function, $u(\phi)$. In order to obtain the dynamic properties of the neural representation (see below) the function must be continuous in ϕ. Therefore, the representation of a particular value of heading direction is a localized distribution of activation with peaks at the specified value (Fig. 12). Such a localized peak of activation is an "instance" of the behavioral dimension heading direction.

The second idea is to generate the neural activation itself from a dynamical system. In other words, the neural activation $u(\phi)$ is a function of time, and evolves continuously in time as determined by a vector field:

$$\dot{u}(\phi, t) = f[u] \quad (5)$$

This makes the function $u(\phi)$ a field, $u(\phi, t)$, in the sense of mathematical physics (that is, continuously many dynamical variables with a topology in the index set, like, for instance, the electric field). If the behavioral variables are sampled discretely we have again a vector-valued ordinary dynamical system. But because the sampling interval is not supposed to matter, this is not conceptually different. For simplicity we use the term neural fields even when sampling is discrete. The brackets in the equation for the neural field dynamics indicate that the vector field at the point $u(\phi, t)$ may depend on the values of the field at all other points $u(\phi', t)$.

Can we define contributions to the neural field dynamics analogously to how we set up contributions to the behavioral dynamics? The ideas are really the same: Each contribution is additive input to the neural dynamics[8]. The contribution is parametrized in

[8] Here again additivity is not a constraint because the functional

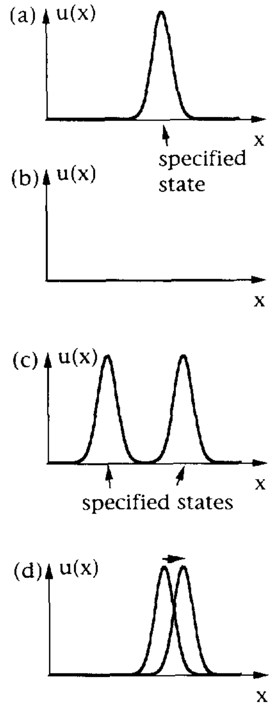


Fig. 12. The concept of a neural field: An activation field, $u(x)$, is defined over a behavioral variable, x. (a) An instance of the behavior is a localized peak of activation, centered on the value of x, which this instance represents. (b) Homogeneous states of activation do not specify any behavior. This limit case, in which there is no instance of the behavioral variable, x, cannot be dealt with in the direct approach to behavioral dynamics, where the behavioral variable, x, is itself governed by a dynamical system. (c) Likewise, multiple instances in the form of multiple localized peaks of activation can exist in the neural field, but not in a direct behavioral dynamics of x. (d) Through strong cooperativity the limit case can be achieved in which peaks move within the neural field as input changes. In this case the neural field emulates a direct behavioral dynamics. If a single peak exists at all times and moves continuously within the field, we speak of a uniquely instantiated system, for which an equivalent direct behavioral dynamics of x can be found.

three ways: (1) Input is local, that is, it is non-zero only in a local region of the space of the behavioral variables. The contents of the input information, that is, which behavior it specifies, determines where the input function is localized. Input may be excitatory (that is, acting towards increasing neural activation) in which case input specifies the values at which it is

form of the input can represent multiplicative interactions by copying the adequate terms.

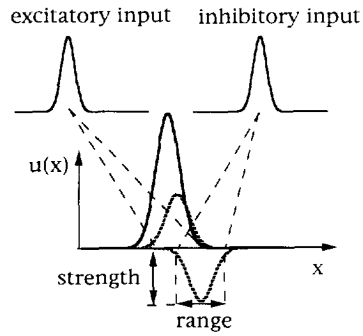


Fig. 13. Contributions to a neural field dynamics can be excitatory or inhibitory. Each contribution is characterized by the location in the field that it specifies, by its strength and by its range. The neural field activation that results (solid line in bottom plot) is not the mere summation of inputs (dashed lines in bottom plot). Strong cooperativity makes the field dynamics very non-linear. As a result, the field may average, but also may make decisions, fuse, create new peaks, delete old peaks, and so on. The same type of analysis as developed for behavioral dynamics can be applied to neural field dynamics.

localized. Or input may be inhibitory (that is, acting towards decreasing neural activation) in which case input specifies the complement of the values at which it is specified. An example of the first is input to the heading directing field from target detectors (Fig. 13). Input from obstacle detectors is inhibitory and localized at the directions in which obstacles are seen. (2) Input functions have a strength, now simply defined as their amplitude. The strength affects how information is integrated with other inputs. (3) Inputs functions have a range, here simply the area in the neural field that receives input from a particular source. Again, the same arguments apply as in the earlier discussion on behavioral dynamics.

The third issue involves interactions within the neural field. One angle from which to approach this is by asking how we might recover within the framework of neural field dynamics the case in which a unique state of the behavioral variables exists at all times and that state changes continuously in time. In other words, how can be obtain the limit case of the behavioral dynamics from this more general picture? This means that the neural field must be able to maintain a localized peak of excitation even in the absence of input information! Thus, the peak must be self-generated by intra-field interactions.

There are actually few mathematical model systems that contain this limit case (see Ref. [36] for a survey). We are sometimes using a set of equations first analyzed by Amari [1] as a model of cortical excitation. In the context of the first example it would read

$$\tau \dot{u}(\phi,t) = -u(\phi,t) + \int w(\phi-\phi')\Theta(u(\phi',t))d\phi' + h + S(\phi,t) \quad (6)$$

wherein τ determines the time scale of the system, w is an interaction kernel typically of mexican-hat form, Θ is a sigmoid threshold function, h a constant determining the overall amount of inhibition or excitation in the system and S represents the potential inputs to this system. In the language of neural modeling, the interaction kernel generates cooperativity. Under certain conditions, only a single peak can be present in the system at any time for any input, because the interaction kernel globally inhibits the rest of the field. At other settings of overall inhibition, h, it is possible, however, to have multiple peaks, which can be induced or deleted by input information (Fig. 12).

In the limit case of global inhibition Amari showed that the self-generated peak, which persists in the absence of input, can be moved around in the field by applying different inputs. If, for instance, no input is present initially, but the peak is present and positioned somewhere, and now an input is applied nearby, then the peak moves continuously from its previous position to a position in which it is centered on the input excitation (see bottom diagram in Fig. 12). Amari provided an equation of motion of this movement of the peak, which could be viewed as the direct derivation of the behavioral dynamics from a neural field dynamics in this limit case of global inhibition. Is it useful to work in this way? Not normally. Clearly, the computational effort of solving an integro-differential equation of the type of Eq. (6) is much larger than that of simply integrating a one-variable differential equation of the type of Eq. (1). However, having established this linkage is important in two respects: First, it shows that the neural field dynamic language is actually more general than the behavioral dynamics language. Second, reversely, it guides us in transferring the concepts of behavioral dynamics to the domain of neural field dynamics: Specification by stabilization, decision making by bifurcations, etc. A discussion of these methods can be found in [22].

We introduce one additional example to illustrate what happens when we apply the neural field ideas to cases that do not correspond to this limit case of behavioral dynamics. Assume that some sensory information provides information about the location of obstacles in the robot's environment. Using ego-position estimation this information can be represented in an allocentric coordinate system. The task is now to represent such obstacle information in memory in a neural field. This problem has been solved in [22] and a report on the implementation of this method on a vision-based autonomous vehicle is given in the next Section . Processes that must be addressed are: the creation of entries in memory, the deletion of entries, the merging or splitting of entries, and the maintenance of entries in the absence of sensory information. The solution has these ingredients: (a) A neural field, $u(x_{obs}, y_{obs})$ represents obstacle locations in the world. The field is operated in the bistable regime where in the absence of input both localized peaks and the homogeneous solution without any localized activation can stably persist. (b) Input information on obstacle locations currently in view excites the neural field where an obstacle is detected and inhibits the neural field where an obstacle is not detected. This leads to the creation of localized peaks at those locations in the field at which no peak existed before but an obstacle is now being detected, and to the deletion of pre-existing peaks where no obstacles are detected anymore. Moreover, if the same area in the world had been seen earlier, and localized activation profiles represented the obstacles at that earlier time, the current sensory information recalibrates these peaks by moving the peaks to the locations at which obstacles are currently being detected. (c) Parts of the neural field which are not currently in the viewing range of the sensors receive no input and hence peaks and homogeneous parts of the field remain stable.

3. Applications

The two applications that are reviewed here both use target acquisition and obstacle avoidance in autonomous vehicles moving over flat ground as the basic reference problem. Because for autonomous vehicles these are the most frequently implemented elementary behaviors, they can serve as reference points from which to assess different approaches. In both cases we solve the basic problem of generating adequate movement based on local sensory information through an attractor dynamics for heading direction. We implement two different, although related, functionalities at a slightly higher level. In the first example (Section 3.1), we show how non-segmented sensor readings (about detected potential obstacle locations) that overlap to varying degrees can be integrated through a dynamic neural field. In the second example, we show how the same type of sensory information can be represented to provide the functionality of memory for obstacles, that is, to provide the capability to use such information after it has stopped being available at the sensors. Memory is subsymbolic and dynamic through the use of a dynamic neural field.

In both cases we are using the same hardware [18] that we refer to as MARVIN (Mobile Active Robot Vehicle for Intelligent Navigation). A commercial robot platform (Robosoft) is equipped with its own, on-board computer system (operating system ALBATROS). A custom-made active stereo-camera system [52] is mounted on the vehicle jointly with an OS9 computer responsible for image-based focusing and camera motion control. This computer also controls the various interfaces. Image acquisition and analysis and the dynamic architecture are implemented on an off-board UNIX computer (Sun 4/330, for some tasks interacting with a Sun SPARCstation 10).

We used a vision based obstacle detection algorithm [57] developed by our colleagues at the Institut für Neuroinformatik in Bochum, which relies on the principle of inverse perspective mapping [35]. The idea is simply to project the images of the two stereo cameras onto the planar surface in which the vehicle moves, which is possible if the camera geometry has been calibrated. Taking the difference between the two images removes all intensity originating from visual structure within the driving surface, because that intensity is identical in left and right view. The remaining intensity comes from visual structure elevated above the driving surface. Any such structure is considered an obstacle. This is a very low-level algorithm, that provides only crude and distorted information about the 3D surface structure giving rise to the image pairs. For instance, a stereo shadow results from the difference in perspective distortion between the two views. This leads to systematic overestimation of obstacle-covered ground over the true footprint of any obstacles.

The obstacle information obtained in this manner is a cloud of points in the driving surface, not segmented and fluctuating from frame to frame. Both dynamic robot architectures integrate this information ultimately in the sense of driving successfully through the terrain without collision. More specifically, dynamic neural representations are generated that reduce the cloud of sensed points to non-cluttered representations within which the vehicle can successfully navigate (see Figs. 16, 18, 19).

3.1. Multisensory integration in a simple obstacle avoidance module

There are two layers of behavioral dynamics in this module (for more details see Ref. [17,19]). Path generation and control is provided by a dynamics of heading direction, ϕ, with contributions representing obstacle and target information. Obstacle information is provided through a neural field dynamics which performs multi-sensory integration.

We use heading direction, $\phi(t)$, in the world as the behavioral variable, because the tasks of target acquisition and obstacle avoidance can be expressed as points in heading direction space and thus heading direction can be made an attractor at all times during motion (as discussed earlier, Section 1). This is heading direction in the world, that is, relative to a world-fixed coordinate axis! The behavioral dynamics is easily designed in this coordinate systems which is why we use it even though orientation in the world is not easily obtained from sensor readings. Actually, only differences between such directions matter. These differences correspond exactly to the measurable directions in which obstacles are seen from the vehicle-fixed view axis. Using constant tangential velocity, v, the robot path is generated by integrating $\dot{x} = v\cos\phi(t)$ and $\dot{y} = v\sin\phi(t)$ where x and y denote the position in world coordinates. We mention these details to illustrate that the concept of "macroscopic" behavior-related variables does not imply that such variables cannot be directly constructed and computed on a given piece of hardware. In the actual implementation all dynamical equations are solved numerically, of course. Discrete increments of position and desired heading direction are computed over a series of steps and then executed by the robot command language[9].

The time series of heading directions is generated by a behavioral dynamics

$$\dot{\phi} = F_{tar}(\phi) + \sum_{obs} F_{obs}(\phi) + \text{noise} \qquad (7)$$

which consists of one contribution representing target acquisition, a sum of contributions representing obstacle avoidance and a gaussian white noise term to ensure escape from unstable fixed points. The target acquisition term

$$F_{tar}(\phi) = -a \sin(\phi - \psi_{tar}) \qquad (8)$$

erects an attractor (in the absence of other contributions) at the direction, ψ_{tar}, in which the target lies from the vehicle (taken relative to a fixed world coordinate orientation). The attractive effect is to extend over the entire range of heading directions. The functional form is dictated by the periodicity requirement for dynamics of angular variables. We have not implemented a sensory module that provides target information, ψ_{tar}. Instead, this part is generated by representing a target in world coordinates and updating ego-position in that world coordinate system through dead-reckoning.

Obstacle information is, by contrast, obtained from the vision module, the inverse perspective mapper. This module delivers a set of points on the driving surface, in vehicle centered coordinates, at which obstacles are detected. Each point is transformed into a polar coordinate system whose orientation is fixed in the world. This orientation is maintained by dead-reckoning heading direction. Thus, input information consists of directions, ψ_{obs}, relative to this world coordinate orientation, and distances, d_{obs}, of points at which the driving surface may be obstructed. In the implementation, this information is obtained from a discretely sampled grid (after several transformations a reflection of the resolution of the video images). The sampling interval on this grid is interpreted as the potential "size" of each obstacle entry.

For the dynamics of heading direction each obstacle contribution erects a repellor at the direction, ψ_{obs}. The functional form of this contribution has been elaborated to deal with finite sizes of obstacles, with distance bias (points close to the vehicle repel more strongly) and assigns a range over which the contribution acts (determined by adjusting the point at which the repellors of two obstacle contributions merge into a single repellor, see below):

$$F_{obs} = \text{Repel}(\phi)\ \text{Range}_{spatial}\ \text{Range}_{ang}(\phi) \qquad (9)$$

with

$$\text{Repel}(\phi) = (\phi - \psi_{obs})/\Delta\psi \times \exp\left[1 - |\phi - \psi_{obst}|/\Delta\psi\right], \qquad (10)$$

$$\text{Range}_{spatial} = \exp\left[-\frac{r_{obs} - R_{obs} - R_{robot}}{d_0}\right], \qquad (11)$$

$$\text{Range}_{ang}(\phi) = \frac{1}{2}[\tanh(h_1(\cos(\phi - \psi_{obst}) - \cos(2\Delta\psi + \delta))) + 1]. \qquad (12)$$

Here $\Delta\psi$ is the angular size of the obstacle, r_{obs} is the distance of the obstacle from the vehicle, R_{obs} and R_{robot} are obstacle and robot radial size, respectively, and h_1, d_0, and $\delta = 0.8$ are model parameters.

The idea here is to look from the vehicle toward the point at which an obstacle has been detected and then imagine copies of the vehicle placed to the left and to the right of that point. The viewing angle, $\Delta\psi$, subtended by this ensemble is the angular range over which the repelling influence is to extend. Thus points nearby will cover a vastly larger angular range than points further away from the vehicle even at constant size of the obstacle element.

Simulation work [48] has demonstrated that this dynamics works as intended, including in situations with dynamic obstacles. Fig. 14 illustrates the basic decision making mechanism in such a simulation. The parameters of the model are calibrated by requiring that the decision not to pass between two obstacles occurs at the adequate spatial distance between the obstacles taking into account vehicle size and safety margins. That critical point is defined as a bifurcation and can be determined by numerical inspection of the dynamics (Fig. 14, bottom) rather than only through trial and error.

Problems arise with this direct approach in cluttered environments such as those defined by the clouds of points coming from the inverse perspective algorithm. To achieve sufficient maneuverability, obstacle

[9] For technical reasons a clothoid method [28] is employed at an intermediate stage to generate an adequate sequence of interpolating robot commands.

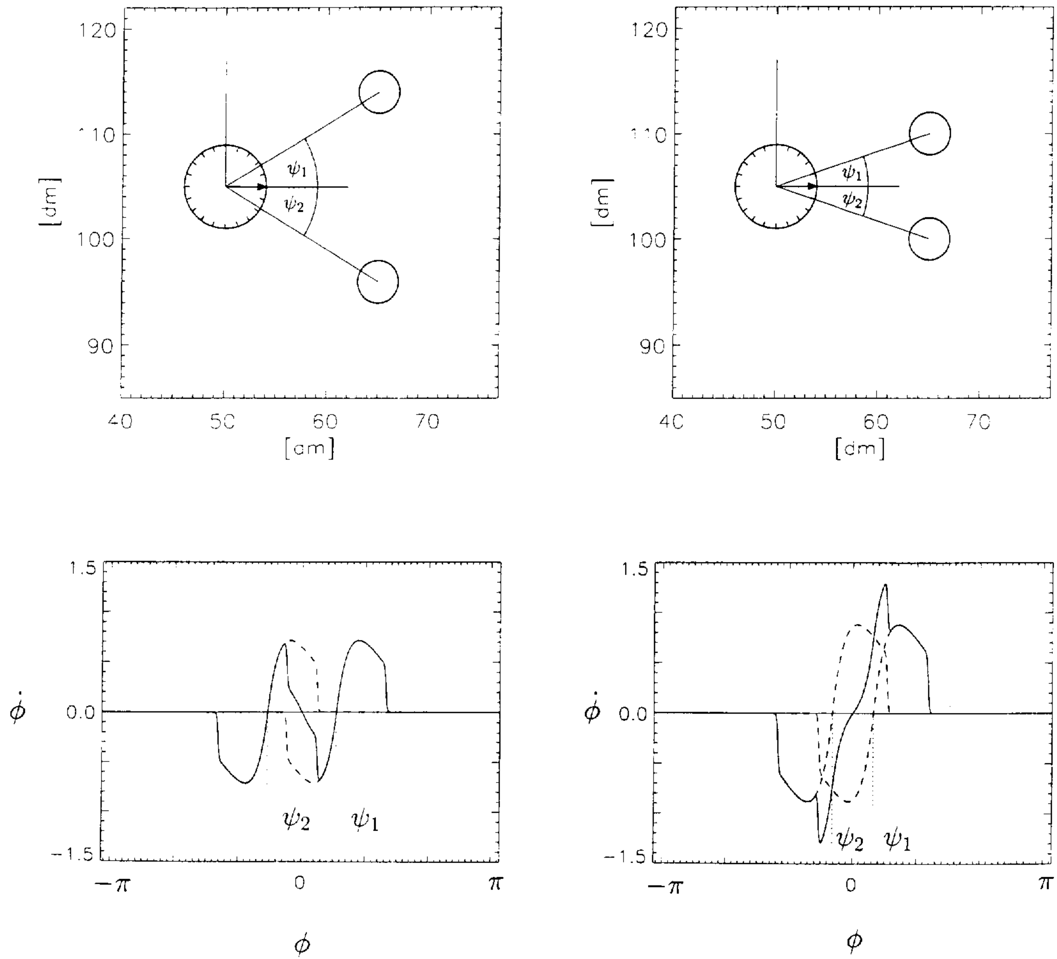


Fig. 14. Calibration of the obstacle avoidance contributions in the first implementation: The minimal integration task involves only two obstacles (small circles on top), the distance between which serves to calibrate the dynamics. When this distance is larger than the vehicle size (large circle on top), passing between the obstacles must be possible (left column). When this distance is smaller than vehicle size (plus some safety margin) then passing must be impossible (right column). The two contributions to the heading direction dynamics (dashed lines in bottom diagrams) are summed (solid lines) and the attractor structure of this sum is examined. For larger than critical distances two repellors are separated by an attractor, for smaller than critical distances a single repellor remains. The parameters of the contributions are tuned such that the bifurcation between these two cases occurs at the distances, at which the vehicle just fits in between the obstacles.

information must be relatively densely sampled. Because each pixel at which obstacle information is detected can lead to a substantial range of heading directions which must be avoided (due to vehicle size and safety margins), the information from the sample sites strongly overlaps when expressed in terms of the contributions to the heading direction dynamics. As a result, directions with multiple detection events may then overpower the contributions of less densely detected signals. Because any repellor function generates an attraction at its boundaries, this leads to spurious attractors that can induce collisions. In potential field approaches this is the well-known spurious minima problem, which occurs generally in additive vector-field approaches (see Ref. [15] for a discussion). The problem is illustrated in a simulation in Fig. 15.

This problem is solved by introducing a new level of behavioral dynamics at which integrated obstacle

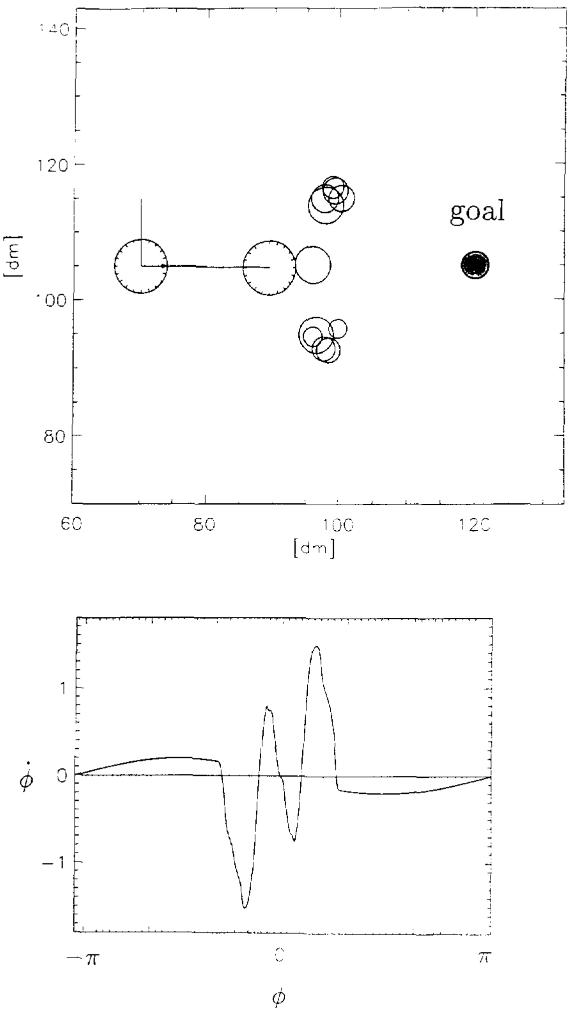


Fig. 15. Spurious attractor problem: Dense or cluttered input which strongly overlaps may lead to spurious attractors. On top, we assume that at two clusters of points obstacles are detected (circles). In addition, a single isolated obstacle is detected (circle in the middle) which blocks the path toward the goal. The dynamic contributions of the two clusters create an attractor for heading between the two clusters, because the top cluster pushes heading to the right, the left cluster pushes heading to the left and the distance between the clusters is sufficiently large to allow passage. The attractor is so strong, that the single repulsive force contribution of the isolated obstacle does not destabilize it. This leads to collision. This simulated situation represents a constructed extremal case of course. In practice, even without a neural field the occurrence of spurious attractors can be minimized by preprocessing obstacle information. In terms of the dynamic approach, the problem is due to the non-local nature of the relevant bifurcation, in which multiple contributions at finite distances from each other matter.

information is represented. The idea is to use a dynamic neural representation that transforms incoming sensory information into a form suitable for path generation. Clusters of detected obstacle information are represented by selected neurons such that when this representation is coupled into the heading direction dynamics then locally the dynamics is similar to the case with two obstacles for which the system has been calibrated (Fig. 14). If this can be achieved, then the calibration based on this elementary situation carries over into general configurations.

We use a neural field in discrete form which is defined on a grid of space points centered in the vehicle. The grid thus moves with the vehicle, but remains in fixed orientation relative to a world coordinate axis (again the exact orientation does not matter but it is important that the grid does not turn with changes in heading direction). This is adequate, because sensory information is to be integrated locally and irrespective of where in the world the vehicle has moved.

Sensory information is projected onto the grid points in two steps. First, the input into the neuronal dynamics at each grid point is determined by looking at all sensor readings that overlap with the sampling radius of the grid site. Because sensor readings are spatially inhomogeneous (here at least due to the inverse perspective, but also, more generally, due to the structure of the visual world) some grid sites integrate much more information than others irrespective of the behavioral relevance of this information. Therefore, the incoming sensory information is added and then thresholded to determine the amount of input excitation. Specifically, input to grid site (i, j) is

$$\mathcal{A}(i,j) = \tanh(c_1 \cdot [-c_3 + \sum_{\substack{\text{sensor}\\ \text{pixels } k}} \Gamma((i,j),k)]) \quad (13)$$

where Γ represents the degree of geometric overlap between sensor pixel and grid site. (Sensor pixels define a small circle centered on the pixel with radius equal to the sampling interval. Similarly grid sites define small circles centered on the grid site with radius equal to the spatial grid sampling interval.) The hyperbolic tangent serves to threshold and normalize raw sensory information. This input is next fed into a competitive activation dynamics

$$\dot{w}_{i,j} = \alpha(i,j)(w_{i,j} - \text{sign}(\alpha(i,j))w_{i,j}^3)$$

$$- \sum_{(r,s)\neq(i,j)} \gamma((i,j),(r,s)) w_{r,s}^2 w_{i,j} + \text{noise} \quad (14)$$

The competitive strength α of each activity variable, $w_{i,j} \in [-1,1]$ is determined both by the strength of input and by spatial distance, closer grid points being stronger than more distant grid points:

$$\alpha(i,j) = (1 + c_2 \cdot \exp(-d/d_\alpha)) \cdot \mathcal{A}(i,j) \quad (15)$$

The degree to which sites compete is determined by $\gamma((i,j),(r,s))$, which decreases as the distance between grid sites increases.

The workings of this neural dynamics can be illustrated with the help of a few limit cases: Sites receiving no sensory information have negative input function and hence negative competitive advantage leading to $w_{i,j} = 0$ as the stable state for those sites. If only a single site receives input within the spatial range of competition, the activated state is stable: $w_{i,j} = \pm 1$. Thus, scattered or isolated input is represented by single, fully activated sites. When sites within the range of competition simultaneously receive input one of two things happens: In the far-field (weak input) one cell is competitively selected, while in the near-field (strong input) more than one cell may be activated, but with a reduced level of activation (see Refs. [17,19] for analytical work). This neural dynamics differs from typical competitive neural activation dynamics primarily through its polynomial functional form, which enables detailed analytical work to determine adequate parameter settings.

Thus the attractor structure of this neural dynamics defines the representation generated at this level. This representation is fed into the heading direction dynamics simply by adding repellor contributions of each grid site, weighted, however, with the level of activation, $|w_{i,j}|$ at that site:

$$\dot{\phi} = F_{\text{tar}}(\phi) + \sum_{\text{grid sites } i,j} |w_{i,j}| F_{i,j} + \text{noise} \quad (16)$$

where $F_{i,j}$ has the functional form of an obstacle contribution for an obstacle located at grid site (i,j) and with a size that reflects the grid constant.

The two levels were implemented through a numerical integration module in the robot control architecture. Other than generating the movement commands by computing upcoming path pieces, the software package developed also permits to monitor the various dynamics by direct on-line graphical representation. In this manner, the adequate parametrization of the dynamics can be checked.

Fig. 16 illustrates how the method works with actual sensory input. In cluttered regions, the neural representation is thinned with a mean distance and strength that conforms to the required elementary situation template. As the vehicle moves, the neural field moves against stationary obstacles. The activated representation then propagates within the neural field as the competitive advantages of the different grid sites change. We have tested this module successfully in numerous runs in the environment of our robot laboratory.

In this application we have integrated multiple sensor readings coming from the same physical sensor, here the stereo range of a stereo camera system (and thus a high-dimensional and inhomogeneous set of sensory information). Similar ideas can also be applied across different sources of behaviorally relevant information. For example, in Ref. [39] the integration of optic flow-based target information with internally represented dead-reckoning information is achieved in a similar manner.

3.2. Dynamic memory in an obstacle avoidance module

This application aims to demonstrate dynamic memory of obstacle information in the same overall task of target acquisition with obstacle avoidance. Memory augments the narrow viewing range ($\pm 27°$) of the stereo camera system and, of course, builds up a representation of the workspace of the robot over time. The methodology of neural dynamic fields is used throughout. Memory is dynamic in the sense that the memory dynamics support the processes of segmenting sensor data and creating memorized instances, as well as updating and deleting such instances, all at a sub-symbolic level of representation. An overt feature of this form of dynamic memory is its capability to cope with dynamically changing environments: whenever sensory information becomes again available the memorized representation is updated to match changes in the environment.

The dynamic neural field architecture is illustrated in Fig. 17. The top layer representing goal position in the world is only simulated. Obstacles are detected in the sense of the inverse perspective mapping the raw

(a)

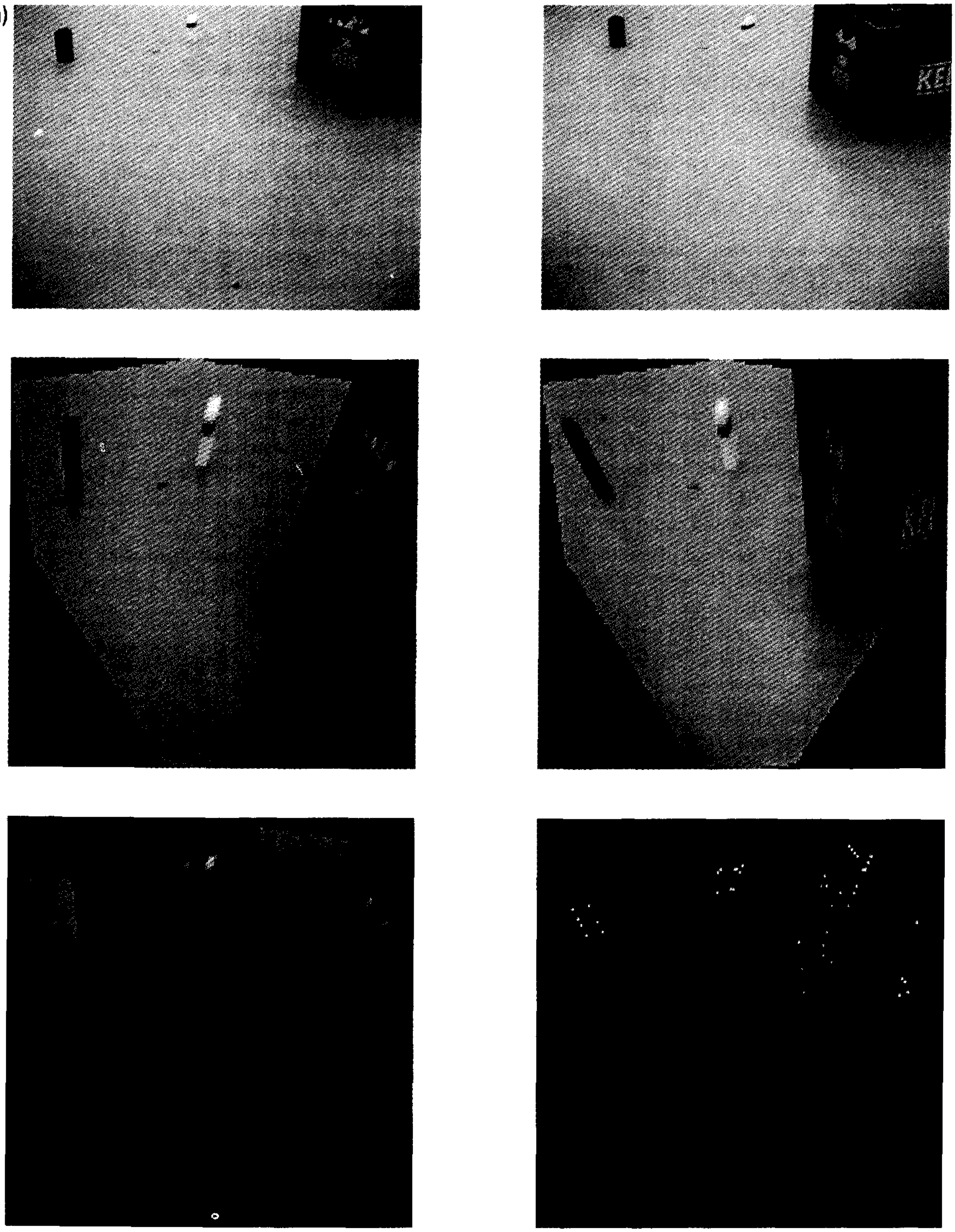


Fig. 16. The implementation of the first application on a vision-based autonomous moving platform. (a) The two images of the stereo camera system (top) are projected onto the driving surface (middle) and subtracted from each other (bottom left). This difference image is thresholded (bottom right) and in this form serves as input into the neural dynamic field.

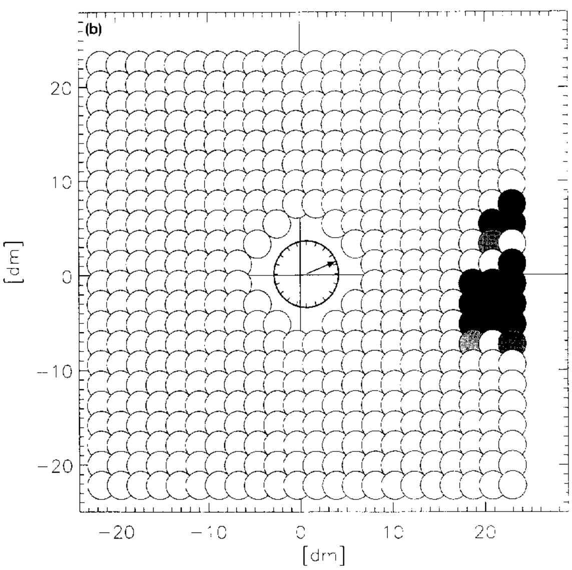


Fig. 16. Continued. (b) The input function field, $\mathcal{A}$, is vehicle centered and performs resampling at desired resolution (circles around grid sites) and normalization.

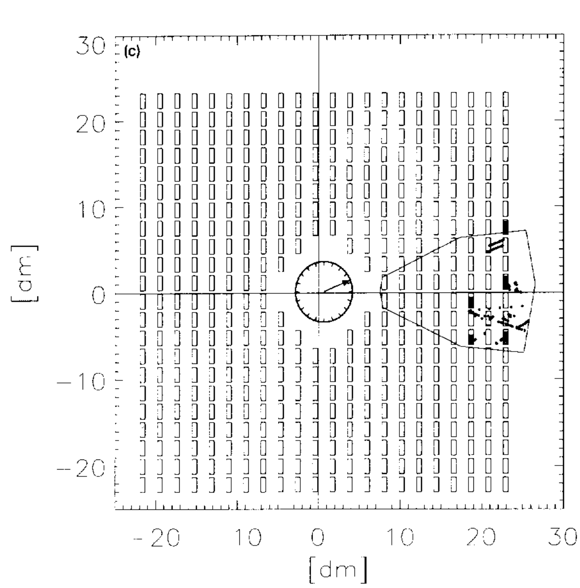


Fig. 16. Continued. (c) The neural dynamic field activates representatives at the grid sites that win a competitive dynamics (filled vertical bars) while all other grid sites have zero activity (open vertical bars) even if they receive input. For illustration the raw input from the inverse perspective mapping is overlaid in this plot (black dots). The polygon indicates the stereo viewing range of the vehicle.

data points of which represent the obstacle layer. The memory and planning layers are dynamic neural fields, mathematically of the Amari type [1]. For an exposition of the mathematics of these fields and detailed simulation work consult [22]. The mathematical formulation is particularly suitable for design because analytic solutions provided by [1] make it possible to relate model parameters directly to desired properties of the solutions. Moreover, the Amari dynamics is particularly suitable as a uniform mathematical formulation that can, in principle, be used across the entire architecture. Thus, the memory layer exemplifies the use of an Amari neural field to represent multiply instantiated information while the planning level exemplifies the limit case in which a unique instance moves continuously within the field so as to emulate the functionality of a uniquely instantiated behavioral dynamics.

The memory field, $u_{\text{mem}}(\boldsymbol{x})$, is defined over the work space, $\boldsymbol{x} = (x_1, x_2)$, of the vehicle in world coordinates. In the implementation, a 10 by 10 meter area is covered. An Amari dynamics

$$\tau_{\text{mem}}\dot{u}_{\text{mem}}(\mathbf{x}, t) = -u_{\text{mem}}(\mathbf{x}, t) + w_{\text{mem}} * \Theta[u_{\text{mem}}](\boldsymbol{x}, t) - h_{\text{mem}} + S_{\text{obs}}(\boldsymbol{x}, t) \qquad (17)$$

operates in the bistable regime without global inhibition. This means the following: Consider first the dynamics without input from S_{obs}. The linear relaxation term, $-u_{\text{mem}}$, the intra-field interaction of the mexican hat type

$$w_{\text{mem}}(\boldsymbol{x}) = \begin{cases} b_{\text{mem}} > 0 & \text{for } |\boldsymbol{x}| < r^{\text{ex}}_{\text{mem}} \\ c_{\text{mem}} < 0 & \text{for } r^{\text{ex}}_{\text{mem}} < |\boldsymbol{x}| < r^{\text{in}}_{\text{mem}} \\ 0 & \text{else} \end{cases} \qquad (18)$$

and the global inhibition constant, h_{mem}, allow for two types of solutions: homogeneous solutions (ϕ-solutions) at low activity and localized solutions (a-solutions) in which peaks of excitation with width a are stable and stationary. The number of localized solutions is limited by the size of the system because peaks keep a minimum distance, which can be determined analytically. The non-linearity generating this multistable characteristic is the threshold function, Θ, which is chosen as a Heaviside step function (0 for negative, 1 for positive argument). The asterisk indicates a two-dimensional convolution $(f * g)(\boldsymbol{x}) = \int d^2\mathbf{x}' f(\mathbf{x} - \mathbf{x}')g(\mathbf{x}')$.

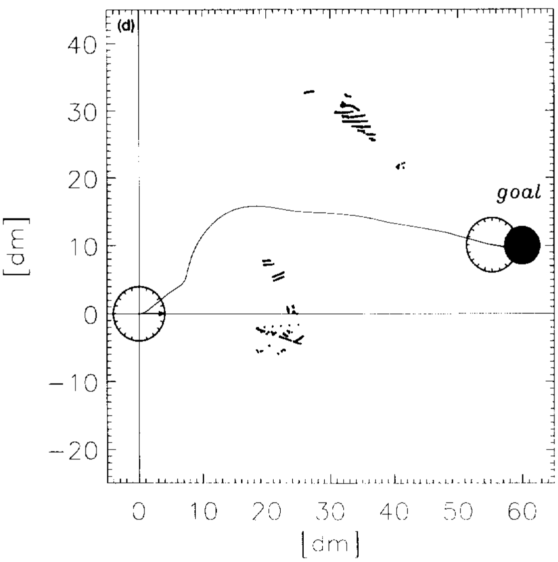


Fig. 16. Continued. (d) The complete two-level dynamics successfully maneuvers the vehicle through an environment to its target (filled circle). The state of the neural fields shown in parts (b) and (c) corresponds to the initial position of the vehicle (left position of the big circle with arrow indicating initial heading direction) The raw input data obtained at various points during the drive are marked by black dots for illustration (the locations in the world at which these data are shown are determined from vehicle position and sensor data).

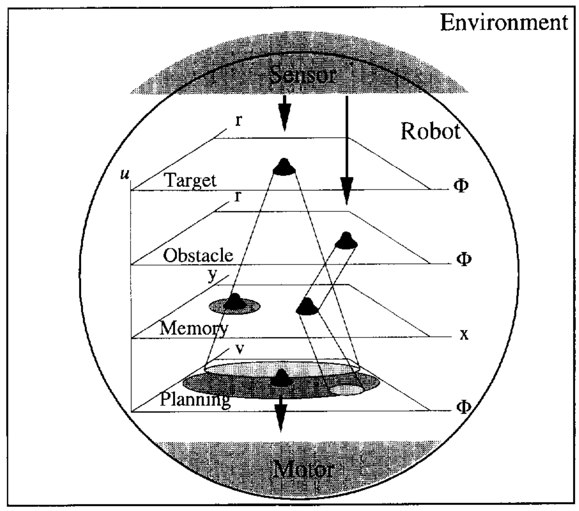


Fig. 17. Neural dynamic field architecture of autonomous vehicle that performs obstacle avoidance and target acquisition with obstacle memory. The target level is only simulated. Obstacles are represented in vehicle-centered polar coordinates (fixed orientation in the world) based on input from the visual inverse perspective mapper. Obstacle memory is a neural dynamic field in world coordinates, into which the obstacle field couples after adequate coordinate transformation (hinted at by oblique projection lines). The memory field couples inhibitorily, the target field excitatorily into the field in which vehicle motion is planned. This field uses heading direction and velocity as dimensions, but we left velocity constant in the actual implementation. Note the broad range of the target contribution.

This field can be operated as a dynamic memory in the following manner: Memory items are individual localized peaks, that is, a-solutions, while the ϕ-solutions represents the absence of information. Excitatory (positive) input from the obstacle field, S_{obs}, creates three types of situations: (1) A transient contribution to input is proportional to the rate of change of sensory information at each location of the viewing range. This component induces a new memory peak if new sensory information arises at a location and deletes a memory peak if sensory information disappears at a location (e.g., by moving). (2) A stationary contribution stabilizes the position of memory peaks at the location specified by current sensory information in the viewing range. This contribution also suppresses localized peaks at locations within the viewing range at which no sensory information is currently being detected. (3) Areas outside the current viewing range receive no input, leaving the currently realized stable state of the memory field invariant.

The nature of this memorized representation of obstacle information is illustrated in Fig. 18 and in the right half of Fig. 19. The incoming sensory information consists of fluctuating clouds of points in space, at which obstacles are detected. The neural field erects a discrete number of peaks that are adequately positioned to cover these clouds. In this way, the environment communicated to the path generation dynamics is cleaned up, so that the elementary decision making process based on a single bifurcation parameter (distance between two obstacles) remains applicable. The memory field thus performs both sensory integration and creation, updating, storing and deleting of memory items.

Note that memory is represented in world coordinates. This makes sense because typically obstacles are expected to be stationary in the world. The sensory information related to obstacle detection is acquired,

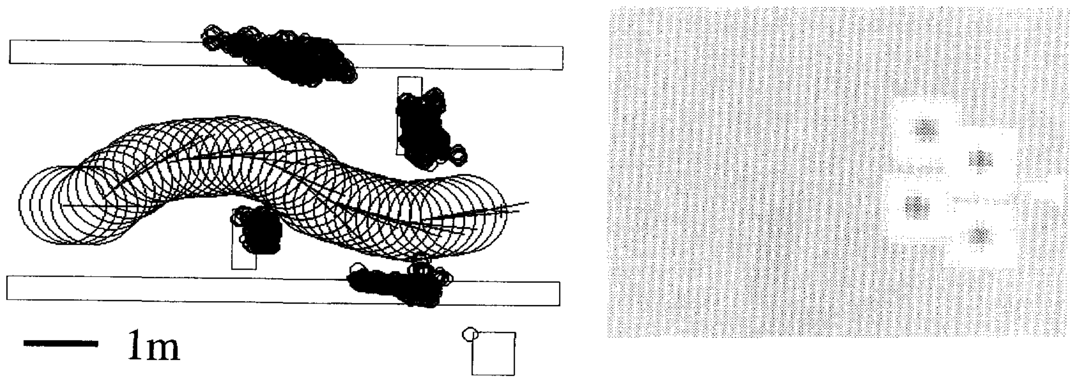


Fig. 18. Obstacle avoidance with memory: We show data from a real run through an environment with obstacles (sketched as rectangles) toward a pre-programmed target to the right of the corridor. The obstacles were cardboard boxes and half- height walls. The small, densely sampled circles indicate detected obstacle information as obtained from the inverse perspective mapping. The information is computed back into world coordinates based on the robot's own dead-reckoning. There are four clusters of detected obstacles due to the directions into which the robot looks as it moves through the parcours. The large circles indicate the subsequent vehicle positions (circle size indicating approximate vehicle size) with the hair pointing in the current heading direction. The memory field is represented to the right as a gray scale image. This field is in world coordinates and covers approximately a 10 by 7.5 m area. Thus the scale of the memory field is smaller in the drawing than the scale of the robot motion. The four points of darker shading are four localized peaks of excitation of the neural field, each matching one of the clusters of detected obstacles. The peaks are surrounded by white areas, in which the field is inhibited more than average (gray background) due to the lateral inhibition. Because only thresholded activation is coupled into the heading direction dynamics, only the darker inner parts of the peaks are significant.

of course, in a coordinate system moving with the vehicle. Thus the coupling of sensory information into the memory field requires a coordinate transformation which involves an estimate of the ego-position of the robot in the world. This estimate is here obtained by integrating the generated path commands, a form of dead reckoning (see below). Thus the architecture as presented in Fig. 17 is not purely feed-forward, even aside from the obvious cooperativity within the field. Instead, the memory field forms part of an internal closed loop.

The path is generated by a dynamics of heading direction, now implemented as a neural dynamic field, $u(\phi, t)$. Its dynamics

$$\begin{aligned}\tau_{\text{path}}\dot{u}_{\text{path}}(\phi, t) &= -u_{\text{path}}(\phi, t) \\ &+ w_{\text{path}} * \Theta_{\text{grad}}(u_{\text{path}})(\phi, t) - h_{\text{path}} \\ &+ f^{\text{inter}}_{\text{path}}[u_{\text{mem}}, u_{\text{target}}](\phi, t) \end{aligned} \tag{19}$$

involves global inhibition, that is, in the kernel w_{path} the inhibitory zone extends over the entire angular range. As a result, the system has a single peak as a stable stationary state and it is around this solution that the system operates. Input from the target and memorized obstacle fields is coupled in positively (excitatorily) for targets and negatively (inhibitorily) for obstacles. These forces position the peak stably at a heading direction that avoids obstacles and heads toward the target. To avoid pinning effects (in which numerical imprecision leads to deviations between the actual and the computed solution) we use a more rounded threshold function, Θ_{grad}. Here convolution takes place in one dimension only. A short-cut eliminates the numerical problem of computing convolutions at this level completely: Using Amari's analytic estimates [1] for the movement of a localized solution under the influence of small inputs we can determine an equation of motion for the position of the peak

$$\begin{aligned}\tau_{\text{path}}\dot{\phi}(t) &= \frac{1}{c}[f^{\text{inter}}_{\text{path}}(\phi(t) + \frac{1}{2}a) \\ &- f^{\text{inter}}_{\text{path}}(\phi(t) - \frac{1}{2}a)]\,, \end{aligned} \tag{20}$$

where a is the width of the localized peaks. As an approximation, this equation can be solved rather than the complete field dynamics Eq. (19). In the implementation on the robot platform we have used this short-cut (for simulations of the full system see Ref. [22].)

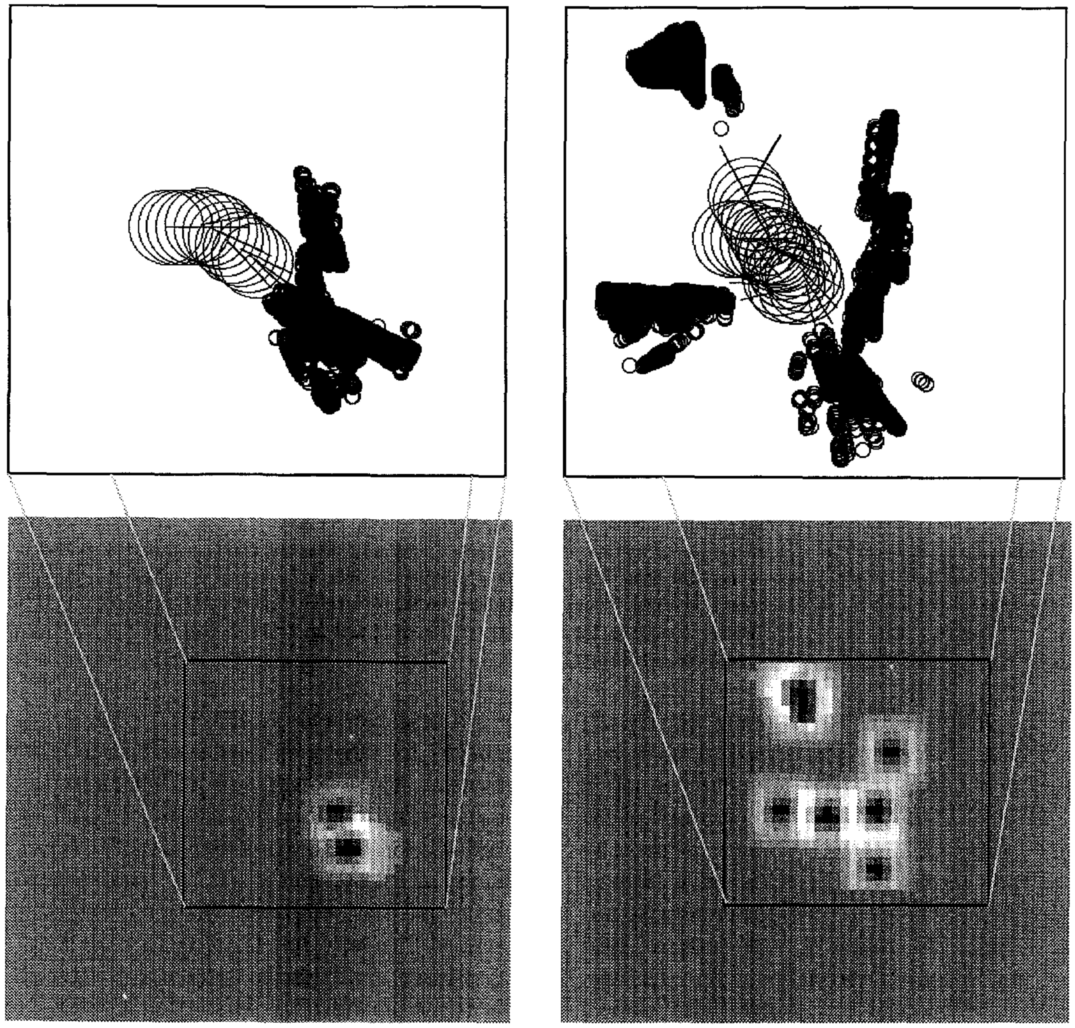


Fig. 19. Obstacle avoidance with and without memory: This figure illustrates the beneficial effect of memory for vision-based navigation with a narrow viewing angle ($\pm 27°$). Again data from real runs are used. The top diagrams show the vehicle position at different time steps (large circles, hair indicating heading direction) jointly with detected obstacle information as provided from the raw inverse perspective mapping (small circles, this information is transformed back into world coordinates for the illustration). In both cases (left and right column) the vehicle was surrounded by the same obstacles. The target was to the right of the initial position. On bottom the obstacle map is shown in world coordinates. The blow-up lines indicate in which part of the world the scene shown on top takes place. White and gray area shadings indicate subthreshold activation, black areas suprathreshold activation. Note the representation of obstacles by isolated localized peaks the distance between which is determined by the intra-field interaction such as to remove spurious attractors from the heading direction dynamics. In the run shown to the left, the memory function was suppressed by eliminating all activation in the memory field outside the current viewing range. Thus, the obstacle representation is limited to a few items detected within the narrow viewing range. As a result, the vehicle oscillates, changing course each time obstacle information shifts as the vehicle starts to turn. The path leads straight to collision with an obstacle. In the run shown to the right, a representation in memory of the obstacles is built up during the initial oscillations. As this representation is augmented, the vehicle starts to turn consistently under the simultaneous influence of this richer representation and ultimately finds the exit from this boxed-in situation to the upper right.

This dynamic architecture was implemented on a UNIX computer in C++ using classes of the Neural Simulation Language NSL developed at the University of Southern California. The program communicates with the sensor module that performs inverse perceptive mapping and with a communications module that passes commands to the robot via UNIX-BSD sockets. Heading direction is sampled at 64 orientations and a 32 by 32 grid is used to discretize the memory field. Based on one acquisition of sensory data the path is integrated to a path length of 20 cm. The target position and orientation are then attained by direct control of two active vehicle wheels, so that the final path is a polygon. Unoptimized code leads to performance in which such a piece of path is computed based on one set of two acquired images in 1.5 sec (including the inverse perspective computation) with an additional 1.5 sec used up for communication with the robot.

Fig. 18 illustrates the functioning of the module. The cluttered incoming sensory information is transformed into a segmented obstacle representation in memory that locally creates a simplified dynamic field for the path generation level, similar as in the previous application. The fact that memorizing obstacle information enhances the obstacle avoidance capabilities of our system is demonstrated in Fig. 19 in which two runs are compared: On the left, movement, sensory information and the dynamic memory field are shown in a situation in which the memory function was artificially suppressed (by resetting the field anywhere outside the current viewing range). Oscillations of the vehicle occur due to the limited viewing range of the obstacle sensor, and lead the vehicle to collide with the obstacles head-on (cf. [33] for a discussion of this problem). With memory (right column) the system easily manages this situation.

4. Discussion

We have employed concepts from dynamics at two levels. The notion of behavioral variables, in terms of which desired and undesired behaviors can be described as points or simple sets, allowed us to talk about behavioral dynamics. The performed behavior is an attractor of these dynamics and the various contributions to the dynamics are all expressed as terms generating either attractors (desired behaviors) or repellors (undesired behaviors). The second step was to introduce dynamics in the form of strongly cooperative interactions within neural representations. In this case, dynamic representations can be generated that are invariant under some classes of change of sensory information. For instance, in the first application, the dynamic representation of obstacles remains unchanged when the incoming sensory information fluctuates or changes systematically as long as these changes in the input are insufficient to change which neurons are winning the competition. Due to hysteresis, the change required to bring about a switch at the neuronal level is finite. In the second implementation the representation of obstacles in a neural field is invariant in this sense: The localized peaks representing entries in memory remain unchanged as sensory information moves through the viewing field and ultimately out of the viewing field as long as this movement is compatible with the visual movement resulting from vehicle motion for an obstacle resting in the world.

There are a number of ways in which the dynamic approach can and must be extended when applications are scaled up from the simplest cases discussed here. We sketch only the most critical issue, the problem of how to organize multiple layers of behavior with the help of time scales. Other aspect such as generating dynamically richer forms of behavior (through limit cycle attractors, or through switching dynamics, etc.) are beyond the scope of this paper.

When the methodology of behavioral dynamics is generalized to multiple layers with multiple behavioral dynamics, an additional ingredient may be needed to provide modularity, that is, to ensure that the individual behaviors can be designed separately. One way to deal with this is invariance. This was used in our applications: One layer was simply invariant under changes in the other layer. In both applications, the representation of obstacles was in a world coordinate system (with respect to orientation in the first case and additionally with respect to the origin of the coordinate system in the second case). As a result, the dynamic representation of obstacles was invariant under changes of heading direction: if we were to rotate the vehicle on the spot, no change in the directions, ψ_{obs}, under which the obstacles are represented would result. Thus, the obstacle representation dynamics could be designed independently of the heading direction dynamics. Reversely, for the heading direction dynamics we treated the representation of the obstacles as quasi-static parameters. This was possible because the obstacle representation had at all times already relaxed to the state representing adequately the sensory input.

This method cannot always be used, however. A new method is needed when different layers of behavior specify particular states for each other. Consider for instance, heading direction as before, but now augmented with a dynamics of velocity, so that the movement of the vehicle is controlled completely by this layer. As a second layer consider a dynamic representation of ego-position in the world integrating various sources of sensory information (in the style used in the second example throughout Section 2). Here, the layer generating movement will directly affect the layer representing ego-position (through dead-reckoning at least, but also through the changed sensory information as the vehicle moves in the world). Conversely, any information taken relative to ego-position (such as the direction towards home if home is stored in world coordinates) now brings about a direct specification for the driving dynamics from the ego-position layer. As a result, the two layers may destabilize or lead to new, spurious fixed points. This would happen, for instance, if heading direction and velocity changed at about the same rate as needed to update ego-position. Because ego-position will determine where the attractors for heading direction lie and heading direction determines where the attractors for ego-position lie, the transients at both layers can either lead to an undesired attractor for both systems (for instance, stopping half way in their relaxation due to a concurrent shift in the attractor of each) or to destabilization (for instance, both variables drifting constantly). How can one deal with such a system?

One attitude that can be taken is, of course, to consider all behavioral degrees of freedom as one big dynamical system, the qualitative dynamics of which is to be analyzed and designed. Not only does this become impractical very quickly, but it may also be inadequate for intrinsic reasons. In the example the problem is that intrinsically an attractor for heading direction may be well-defined only given a particular estimate of ego-position (for instance, for a contribution leading to returning toward a memorized target position represented in allocentric coordinates). Reversely, the attractor for ego-position can be computed only given the current motion command (if dead-reckoning is to be used). Thus, one cannot design the behavioral dynamics adequately without separating these two layers into two different behavioral dimensions.

The cue to these sorts of problems is the concept of time scale: the time over which individual components of the behavioral dynamics relax. Close to fixed points these time scales can be analyzed in terms of the real parts of the eigenvalues of the local stability problem (similar as in control theory). Dimensions along which the system relaxes with different time scale have clear-cut dynamic relationships: the slower directions control the faster directions. Intuitively speaking, the faster variables relax to the value prescribed by the slower variables before these latter change appreciably. Thus, from the point of view of the faster variables the slower ones are quasi-constant parameters (one says, they are adiabatic variables). From the point of view of the slower variables, the faster ones can be eliminated by assuming they have already relaxed to the adequate attractor (one speaks of adiabatic elimination). Thus, given different time scales, the two layers can be designed separately.

The more general idea is thus to cut the high dimensional state space of a multi-layered system into individual layers by introducing differences in time scale among different variables. Then the behavioral dynamics at the various levels may be analyzed and designed separately using such techniques as adiabatic elimination or multi-scaling. (A mathematical treatment of these sorts of problems is provided, for instance, in [30] in terms of multi-scaling methods and in [41] in terms of the Center Manifold Theorem. Adiabatic elimination is treated along more intuitive lines in [24].)

Does this lead to a simple fixed hierarchy of behaviors, ordered along increasing time scales, with the slowest variables on top? The answer is no: such a hierarchy may exist at a given moment of time, but it may change in time as a result of the ongoing dynamics itself. Such changes are brought about again by instabilities. When a dynamics in whatever layer goes through an instabilities it becomes momentarily very slow. Thus, in effect, such a layer moves to the top of the hierarchy while it is near the instability. In a sense, the behavioral dimensions relative to which the system is currently going through a critical "decision" point are governing all other variables. The resulting hierarchy is thus highly flexible and, yes, dynamic (see also part (3) in Section 4.2).

4.1. Relation to other work

Behavior-based approach. As hinted in the Introduction, the dynamic approach has a lot in common with the behavior-based approach to autonomous robots [9,12,42], the two most important points being: (1) The system is designed in terms of elementary behaviors. (2) Sensory information is used at a low level of processing and in a manner specific to the behavior that this information feeds into. The dynamic approach diverges in several aspects, however. First, the emphasis on theory in the dynamic approach contrasts with the rather atheoretical (or even anti-theoretical) emphasis in some of the behavior-based work. This means, in particular, that in the dynamic approach the behavior of the robot is represented internally (through the behavioral variables). In fact, the main trick of the approach is to work with these behavioral variables that isomorphically represent the overt behavior of the robot. By contrast, behavior-based approaches stress the concept of directly acting out behavior. A second obvious difference is that in the dynamic approach representations of information are explicitly used. The philosophy has been to treat representations in the same way as overt behaviors. This did not prevent us from dealing both with external information and with the representations of internal states of the system in world coordinates, for instance.

Like subsumption [11] or schema theory [3] an attempt is made to provide a uniform language in which architectures for autonomous systems can be formulated. The main distinctive feature of the proposed language is the concept of (asymptotic) stability, through which both control stability and decision making/flexibility is brought about. This reaches into the use of neural representations, which are structured through the same type of analysis (see also below).

It may be useful to look at dynamic architectures to see in which way behaviors may interact and compare this to the goals and realities of behavior-based architectures. There are really two types of interactions within dynamic architectures: (1) A given layer of behavior can specify states at another layer of behavior. This means that the given layer determines the position of attractors or repellors of the dynamics of the other layer. For instance, in our applications the obstacle representation acts onto the heading direction dynamics in this way. From this form of interaction various different kinds of functionalities can be built such as sensory fusion, information processing, and both feed-forward and feed-back stabilized movement behavior. (2) On the other hand, a given behavior can activate or deactivate another behavior. This occurs through the strength of the input of one layer into the other and would typically involve a neural representation. This type of interaction occurs, for instance, within an individual neural field when isolated instances are created as separate localized activity peaks. More generally, entire behaviors could be activated or deactivated in this manner, so that subsystems can be taken in and out of the global perception-action stream. In the jargon of subsumption, the first type of interaction might be similar to subsuming itself, the second to inhibiting output or suppressing input.

Potential field approach. The dynamic approach shares with potential field methods [2,26,34,31,59] the idea of integrating multiple constraints by adding contributions at the level of a vector field. The main difference is that in the dynamical approach the system is at all times in an attractor during operation, while in the potential field approach behavior is planned by following a transient solution. To see this consider again obstacle avoidance. In common potential field algorithms the spatial position of the vehicle is the variable over which a vector field is defined. The target is made attractive (minimum of a potential) and obstacles are made repulsive (maxima of a potential). Paths are generated by moving downhill in the potential using various types of algorithms. In terms of dynamics this corresponds to the process of relaxation toward the attractor at the target. Thus the actual behavior is the transient. The only moment in time when the system is in an attractor is after it has reached the goal. (So, in dynamic terms, these models essentially generate postural behavior.) One practical implication is that the design of the system, that is, determining the functional forms of the various contributions to the potential, cannot be based on an analysis of the attractor structure of the dynamics and its bifurcations. The transient solutions are, in fact, not systematically affected by instabilities that occur somewhere in the vector field.

This difference may seem to be of a rather technical nature, but it is not. One conceptual consequence is that the dynamic approach contributes to the control-theoretic stability of the overall behavior as it is acted out in the world while potential field approaches do not. This is why using the dynamic approach pushes us to consider the closed perception-action loop of the system and puts us closer to behavior-based methods, while the potential field approach actually originated as a heuristic method of planning and is not typically considered as part of the closed loop. A second important consequence of this difference is that the dynamic approach makes specific demands on the types of variable used. Only for specific choices of the behavioral variables is it possible to have the system behave while sitting in an attractor. In the example of obstacle avoidance, for instance, using the cartesian position of the robot as the variable in terms of which the system plans and generates behavior, precludes this possibility: The only behavior that corresponds to an attractor of this variable is posture, resting where the system currently is.

Finally, we have shown how the construction of layers of dynamic representations can successfully deal with the notorious problem of spurious states from which potential field approaches suffer, without giving up the locality and closed-loop characteristic of the method.

Other dynamic approaches. Concepts from the theory of dynamical systems begin to be applied by other researchers to autonomous robotics (e.g., [6,55,53]), and more widely, to autonomy in a more abstract sense [7]. Randall Beer [6] has sketched a general framework somewhat similar to ours. The primary difference is that he uses dynamics as a method of analysis (of systems generated by evolutionary methods), while we are using dynamics as a method of design. A conceptual difference is that in our approach the nature of the behavioral variables is quite explicitly constrained by the requirement that goals can be expressed as values or sets of values of these variables. As a consequence, we distinguish between behavioral variables and neural activation. Neural activation in our approach also takes the form of dynamical variables, but these have a different significance and couple differently into other layers than behavioral variables (that is, they couple in terms of strength of information, not in terms of specifying behavior). By contrast, in Beer's framework, the nature of the dynamical variables is left quite open. A third difference is again, as relative to the potential field approach, our principle of generating behavior through attractors. Beer employs transient trajectories to embody behavior.

Tim Smithers [53] is arguing for the use of dynamical systems to describe the agent-environment relationship. His distinction between interaction and infrastructure dynamics is particularly useful. The shared emphasis is on principles of design, which enable rational choice of system parameters and architectures. Again, the primary difference with our approach is our insistence on specific prescriptions of how to identify adequate variables and the mapping of behaviors onto attractors. Somewhat related ideas have also been expressed by Luc Steels [55], although the analogy with dynamics does not seem to go beyond metaphor at the moment.

Neural network theory. The concept of a neural field dynamics has precursors in the domain of modeling target selecting and binocular function in nervous systems [27,13] although the aspect of self-generation of localized peaks was not stressed in that work. More generally, the principle of neural representation is, of course, widely used in the field of neural network modeling. To avoid confusion we stress that we have not addressed issues of learning in neural representations at all, although the framework offers this avenue. Work on the self-organization of motor-maps in the context of conventional robotics [44] leads the way in this direction. Arguments for topological representations for autonomous robots are given in [23,61] (see also references therein).

The dynamic approach brings in two elements that are to some extent new in this domain: (1) By linking neural fields to the concept of behavioral dynamics we provide new criteria with which to design neural networks. In particular, the distinction between specifying information and strength of information, and the concept of range of specification were derived by establishing behavioral dynamics as a limit case of neural field dynamics. Linking decision making to bifurcations allows the rational design of neural interactions, for instance, in the context of sensory fusion.

(2) The principle of self-generation of activation patterns is a new emphasis in this domain. Under adequate circumstances, intra-field interactions dominate over input, which serves only to position a marginally stable localized activation peak. This makes it possible to generate memory representations, and to realize systems, in which a unique instance exists at all times, independently of the presence of input. More abstractly, it is through self-generation of neural activation that the neural representations we have developed attain their invariance properties. In a sense, this is a possible mechanism for symbol grounding.

Grid methods of sensory fusion. A more specific relationship of our approach to established methods invites comment. In the domain of sensory fusion, methods have been developed in which sensory information is accumulated to generate incrementally a world model (see [20] for review, see also [8]). These methods closely resemble our neural representations, for instance, of obstacles in the world. The basic conceptual difference is that our representation is itself a behavior, that is, it has dynamic stability properties and is structured by intrinsic constraints, not only driven by input information. To exemplify this somewhat abstract point we explain a subtle technical difference: The neural representations of obstacle position in our two applications are structured by the demands of the heading direction dynamics, into which these representations feed. In particular, the problem of spurious attractors of a heading direction dynamics in the presence of cluttered sensory information was addressed by designing the neural representations such that a particular minimal distance between activated representatives of obstacle information was achieved. For instance, in the Amari neural field dynamics, this distance is visible as the separation of localized peaks of activation (Figs. 18, 19) and is determined by the size of the mexican hat interaction kernel). This minimal distance between representatives is not, however, the spatial resolution of the obstacle representation. The resolution is given by the sampling of the neural field or the grid size of the discrete neural network. Thus, for instance, an ensemble of spatially separated peaks may be shifted in the plane by amounts much smaller than the minimal separation of peaks. This is what happens continuously while the neural representation is coupled to incoming information. By contrast, in the grid histogram techniques, the resolutions of the resulting world representation is set by the grid size. If in such representations an attempt were to be made to reduce the clutter of sensory information by generating a minimal distance between entries it would have to come at considerable cost to spatial resolution.

The fact that neural representations are designed like behaviors based on task requirements and with their intrinsic dynamic properties is crucial for the integration of such representations into complete system architecture. Both the uniformity of the theoretical concepts and, more specifically, the existence of a time scale in the sense of dynamics, are prerequisites for such integration.

4.2. Strengths

The primary strength of the dynamic approach might be its degree of theoretical penetration. By mapping behavior onto attractors and expressing behavioral constraints as contributions to the dynamics that define either attractors or repellors, the mathematics of the qualitative theory of dynamical systems become tools for the design, analysis and maintenance of autonomous robot systems. Conceptually, there are two important implications: (1) Through behavioral dynamics those functionalities that are typically described in terms of planning, representation, and information processing are endowed with dynamic stability properties and can thus be dealt with in a manner that is consistent with prevalent control theoretic concerns. (2) By making sensory information define attractors or repellors of behavioral dynamics much of the computational burden is lifted off sensory information processing. All that is needed for a sensor is that it provides information about tendencies of desirable or undesirable behavior, or, in other words, that it "specifies behavior". Specifying behavior does not necessarily mean that all parameters of the behavior must be computed. Any structured contribution capable of indicating a direction in the phase space of the behavior dynamics in which to move can make a contribution.

A few concrete implications can be discussed that flow from these conceptual issues: (1) Decision making occurs in the sense that small and unspecific changes in sensory information can lead to qualitative change of behavior. Such changes are brought about by instabilities. Bifurcation theory can be used to design a behavioral dynamics such that it goes through a desired bifurcation under the adequate conditions. We illustrated, how the decision whether or not to pass through between two obstacles can be designed by constructing the pitchfork bifurcation that links the two limit cases (of passing and of going around) and adjusting the parameters of the dynamics so that the bifurcation occurs at the right point (at the distance between obstacles at which the vehicle just fits in between).

(2) Sensory fusion involves a related use of instabilities. When multiple contribution to a behavioral dynamics overlap to varying degrees with respect to the behavior they specify, the corresponding contributions to the behavioral dynamics may either essentially superpose leading to forms of averaging among the behaviors specified by each sensory channel. Alternatively, if discrepancies between the behaviors specified by such contributions are excessive, a bifurcation leads to a splitting up of the resultant attractor. We illustrated how a pitchfork bifurcation can lead to two separate attractors, each relying on a different sensory channel. Which attractor the sensory representation relaxes to then depends on the recent history of the system. An alternative scenario removes one of the attractors through a tangent bifurcation. The control over which scenario occurs and at which point this bifurcations take place is again in the hands of the designer, who can set parameters relative to strength and range of the contributions to the dynamics.

(3) Because each behavioral module is a dynamics with attractor solutions, its intrinsic time scale can be assessed in terms of the relaxation time of its attractors. System integration is then the problem of coupling different such dynamics. The various techniques for dealing with dynamics that differ in time scale (that is, in relaxation time) can thus be used to design in which manner these behavioral modules will control each other. Essentially, slower modules always control faster modules (slow and fast always in terms of time to relax to the corresponding attractor state). When a particular module must be dominant the designer can assign a slow relaxation time to the module. However, because relaxation times change as the qualitative dynamics changes, which module in effect dominates may depend on the behavioral situation. By performing the adequate bifurcation analysis, the designer can determine under which circumstances which module is to be the dominant one. For a simple example, consider ego-position estimation based on dead-reckoning and on landmarks. Usually, dead-reckoning will be the slow variable dominating the integrated ego-position estimate. When a landmark is encountered, the integrated ego-position system can be made slow, so that it now dominates and resets the dead-reckoning system [56]. Needless to say, these are only the simplest of manifold possibilities through which behavioral systems can be structured by adjusting its interaction dynamics. Much remains to be explored in this direction.

(4) Examining the effective behavioral dynamics may also be a way in which the engineer can determine to which degree the often neglected "infrastructure dynamics" [53] is consistent with the design goals. By design, sensory information is assumed to specify behavior. When in implementation the dynamics generated by on-line sensory information is analyzed in terms of the resultant attractors and their stabilities the quality of such sensory information (and the validity of the design assumptions) can be assessed. At the other end, by comparing the behavioral state as expressed in the behavioral dynamics with the real-time effector movement the degree to which the infrastructure dynamics links uniquely the behavioral level to the effector level can be diagnosed. Adjusting the time scale of the behavioral dynamics so that it is slower than the infrastructure dynamics is the simplest design option to eliminate problems at this end.

(5) For higher cognitive or sensory functions the principle of neural representation is essential. Drawing on recent progress in neural network theory the feedforward structure of neural fields can be designed to perform desired transformations. The key element, self-generation of neural activation, adds invariance of representation under change of sensory information to the tool kit. In our examples, for instance, we provided a simple form of obstacle position memory which was invariant under the transformations of sensory information brought about by the vehicle moving in the world.

4.3. Drawbacks

The major drawback of the dynamic approach is its computational cost. This is a serious issue only when neural dynamic fields are used. These integro-differential equations must be solved with sufficient spatial resolution to avoid pinning effects, in which localized solutions get stuck on the grid points. Because the equations involve convolutions over the field, the computational cost increases quickly with the dimensionality of the problem. This drawback is not limited to implementation on serial computers because the strongly cooperative nature of the dynamics creates potential problems for parallel implementation as well.

Other than by waiting for the famed "brain-like" computing devices (which obviously would be able to do this just as our cortex does) this problem can be addressed in two ways: (1) Designing behaviors in highly modular fashion limits the maximal dimensionality of fields. Indeed, once this has been recognized as a problem, one can achieve a lot while never going beyond one-dimensional fields. (2) More radically, one might view the dynamic approach merely as the theoretical framework from which to structure a design. The actual implementation might then use much simpler mathematical techniques, for instance, algorithms, that mimic the dynamic properties needed from the theoretical point of view. For instance, sensory fusion with bifurcations leads essentially to hysteretic decisions about which channel to follow. This might be directly implementable in an algorithm that does not actually solve the neural field dynamics but instead reads out maxima of the input, for instance, and keeps a single dynamical variable responsible for generating hysteresis effects to stabilize the decision. A mild form of this attitude was the "short-cut" through which we transformed neural fields into computationally trivial ordinary differential equations for the uniquely instantiated parameter heading direction (second application).

4.4. Perspectives

To us there seem to be two directions in which progress can be made immediately by developing the concepts discussed here. One is by carrying the work toward higher cognitive functions. The issue to be addressed is that of invariance, in which active behavior may allow an autonomous system to extract behavior-specific information even as the information received at the sensory surface varies. First steps in this direction were made here. The second direction is to more seriously probe the practical usefulness of the dynamic approach by developing design, analysis and diagnostic tools in real robot environments. The goal would be to make an autonomous robot truly predictable, and thus robust and manageable.

More globally, part of the appeal of the dynamics approach is its unifying nature, drawing together aspects from various subfields. For instance, the potential field planning methodology is brought in line with control theoretic concepts, and these are combined with neural network ideas. The full potential of this unification has not yet been realized at all. For instance, integrating learning into dynamical architectures now seems possible and desirable (a candidate system is [43]). The relationship to fuzzy control can be explored and is potentially fruitful (see [21] for a discussion).

Acknowledgements

Work supported through a grant from the BMFT, Germany within the NAMOS program. We thank Stefan Fuhrmann, Thomas Bergener and Ekkehard Schulze-Krüger for technical support. Discussions with Randall Beer, Wolfram Erlhagen, Klaus Kopecz, Hartmut Neven, Rolf Pfeifer, and Tim Smithers are gratefully acknowledged. An anonymous referee made helpful detailed comments that pushed us to much simplify and (hopefully) clarify the paper.

References

[1] S Amari, Dynamics of pattern formation in lateral-inhibition type neural fields, *Biological Cybernetics*, 27:77–87, 1977.

[2] J.R. Andrews and N Hogan, Impedance control as a framework for implementing obstacle avoidance in a manipulator, In E.E. Hardt and W Book, editors, *Control of Manufacturing Processes and Robotic Systems*, pages 243–251. ASME, Boston, 1983.

[3] M.A. Arbib, Visuomotor coordination: Neural models and perceptual models, In J.P. Ewert and M.A. Arbib, editors, *Visuomotor Coordination*, pages 121–174, New York, London, 1989. Plenum Press.

[4] R.C. Arkin, Motor schema-based mobile robot navigation, *Int. J. Robotics Research*, pages 92–112, 1989.

[5] R.C. Arkin, Integrating behavioral, perceptual, and world knowledge in reactive navigation, *Robotics and autonomous control*, 6:105–122, 1990.

[6] R.D. Beer, A dynamical systems perspective on agent-environment interaction, *Artificial Intelligence* 72:173–215, 1995

[7] M.H. Bickhard, Representational content in humans and mashines, *J. Experimental and Artificial Intelligence*, 5:285–333, 1993.

[8] J. Borenstein and Y. Koren, Real-time obstacle avoidance for fast mobile robots in cluttered environments, In *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 572–577, 1990.

[9] V. Braitenberg, *Vehicles. Experiments in Synthetic Psychology*, MIT Press, Cambridge, Mass., 1984.

[10] M. Braun, *Differential equations and their applications*, Springer Verlag, New York, 1978.

[11] R.A. Brooks, A robust layered control system for a mobile robot, *IEEE Journal of Robotics and Automation*, RA-2:12–23, 1986.

[12] R.A. Brooks, New approches to robotics, *Science*, 253:1227–1232, 1991.

[13] R. Chipalkatti and M.A. Arbib, The cue interaction model of depth perception: a stability analysis, *Journal of Mathematical Biology*, 26:235–262, 1988.

[14] J.H. Connell, *A colony architecture for an artificial creature*, PhD thesis, MIT Electrical Engineering and Computer Science, 1988, MIT AI Lab Tech Report 1151.

[15] C.I. Connolly, J.B. Burns, and R. Weiss, Path planning using Lapalce's equation, *IEEE Robotics and Automation*, pages 2102–2106, 1990.

[16] F. Delcomyn, Perturbation of the motor system in freely walking cockroaches. i. rear leg amputation and the timing of motor activity in leg muscles, *Journal of experimental Biology*, 156:483–502, 1991.

[17] M. Dose, *Wegeplanung autonomer mobiler Roboter mittels dynamischer Systeme*, PhD thesis, Universität Dortmund, Abteilung Informatik, 1994.

[18] M. Dose, S. Fuhrmann, E. Schulze-Krüger, and W.M. Theimer, An autonomous mobile robot: a development tool for operation in a natural environment, Technical report, ISSN 0943-2752, Institut für Neuroinformatik, Ruhr-Universität Bochum, Germany, 1994.

[19] M. Dose and G. Schöner, Implementing a dynamic field architecture on a vision-based mobile robot platform to achieve sensory integration, VDI-Verlag, Germany, in press.

[20] A. Elfes, Sonar-based real-world mapping and navigation, *IEEE J. Robotics and Automation*, RA-3:249–265, 1987.

[21] C. Engels and G. Schöner, Dynamic neural field architecture for the design of autonomous systems, In E. Köhler, editor, *39. Internationales Wissenschaftliches Kolloquium*, pages 47–52. Technische Universität Ilmenau, 1994.

[22] C. Engels and G. Schöner, Dynamic fields endow behavior-based robots with representations, *Robotics and autonomous systems*, 14:55–77, 1995.

[23] P. Gaussier and S. Zrehen, Complex neural architectures for emerging cognitive abilities in an autonomous system, In P. Gaussier and J-D Nicoud, editors, *Proceedings From Perception to Action Conference, September 7-9, 1994, Lausanne, Switzerland*, pages 278–289, Los Alamitos, California, 1994. IEEE Computer Society Press.

[24] H. Haken, *Synergetics–An Introduction*, Springer Verlag, Berlin, 3 edition, 1983.

[25] H.S. Hock, J.A.S. Kelso, and G. Schöner, Perceptual stability in the perceptual organization of apparent motion patterns, *Journal of Experimental Psychology: Human Perception and Performance*, 19:63–80, 1993.

[26] N. Hogan, Control strategies for complex movements derived from physical systems theory, In H. Haken, editor, *Complex Systems–Operational Approaches*, pages 156–168. Springer Verlag, Berlin, 1985.

[27] D.H. House, A model of the visual localization of prey by frog and toad, *Biological Cybernetics*, 58:173–192, 1988.

[28] Y. Kanayama and N. Miyake, Trajectory generation for mobile robots, *Robotics Research*, 3:333–340, 1986.

[29] J.A.S. Kelso, M. Ding, and Schöner G, Dynamic pattern formation: A tutorial, In Smith L B and Thelen E, editors, *A dynamic systems approach to Developement: Applications*, pages 13–50. The MIT Press, Boston, 1993.

[30] J. Kerkovian and J.D. Cole, *Perturbation Methods in Applied Mathematics*, Springer Verlag, New York, 1991.

[31] O. Khatib, Real-time obstacle avoidance for manipulators and mobile robots, *Int. J. Robotics Research*, 5:90–98, 1986.

[32] K. Kopecz and G. Schöner, Saccadic motor planning by integrating visual information and pre-information on neural, dynamic fields, *Biological Cybernetics*, 73:49–60, 1995.

[33] Y. Koren and J. Borenstein, Potential field methods and their inherent limitations for mobile robot navigation, In *Proceedings of the 1991 IEEE International Conference on Robotics and Automation*, pages 1398–1404, Sacramento, California, 1991.

[34] B.H. Krogh and C.E. Thorpe, Integrated path planning and dynamic steering control for autonomous vehicles, In *Proc. 1986 IEEE Int. Conf. on Robotics and Automation*, pp. 1664–1669, 1986.

[35] H. Mallot, H Bülthoff, J.J. Little, and S. Bohrer, Inverse perspective mapping simplifies optical flow computation and obstacle detection, *Biological Cybernetics*, 64:172–185, 1991.

[36] A.S. Mikhailov, *Foundations of Synergetics I. Distributed active Systems*, Springer, Berlin, 1990.

[37] H. Mittelstaedt, Basic solutions to the problem of head-centric visual localization, In R. Warren and A.H. Wertheim, editors, *The Perception and Control of Self-Motion*, Hillsdale, New Jersey, 1990. Erlbaum.

[38] W.L. Nelson and I.J. Cox, Local path control for an autonomous vehicle, In *Proc. 1988 IEEE Int. Conf. on Robotics and Automation, Philadelphia, Pennsylvania, April 24-29, 1988*, pages 1504–1510, 1988.

[39] H. Neven and G. Schöner, Neural dynamics parametrically controlled by image correlations organize robot navigation, submitted.

[40] D. W. Payton, Internalized plans: A representation for action resources, *Robotics and Autonomous Systems*, 6:89–103, 1990.

[41] L. Perko, *Differential Equations and Dynamical Systems*, Springer Verlag, Berlin, 1991.

[42] R. Pfeifer and C. Scheier, From perception to action: The right direction? In P. Gaussier and J-D Nicoud,

editors, *Proceedings From Perception to Action Conference, Lausanne Sept 1994*, pages 1-23, Los Alamitos, CA, 1994. IEEE Computer Society Press.

[43] Rolf Pfeifer and Paul F M.J. Verschure, Designing efficiently navigating non-goal-directed robots, In Jean-Arcady Meyer and Stewart W. Wilson, editors, *From Animals to Animats: The second international Conference on Simulation of Adaptive Behavior*, pages 31-39, Cambridge, MA, 1993. MIT Press.

[44] H. Ritter, T. Martinetz, and K. Schulten, *Neuronale Netze*, Addsion-Wesley, 2 edition, 1991.

[45] H. Ritter, T. Martinetz, and T. Schulten, Topology conserving maps for learning visuomotor-coordination, *Neural Networks*, 2:159-168, 1989.

[46] G. Schöner, Dynamic theory of action-perception patterns: the "moving room" paradigm, *Biological Cybernetics*, 64:455-462, 1991.

[47] G. Schöner, From interlimb coordination to trajectory formation: common dynamical principles, In S. Swinnen, H. Heuer, J. Massion, and P. Casaer, editors, *Interlimb coordination: neural, dynamical, and cognitive constraints*, pages 339-368. Academic Press, San Diego, 1994.

[48] G. Schöner and M. Dose, A dynamical systems approach to task-level system integration used to plan and control autonomous vehicle motion, *Robotics and Autonomous Systems*, 10:253-267, 1992.

[49] G. Schöner and W. Erlhagen, A dynamic neural field theory of motor programming, submitted.

[50] G. Schöner and J A.S. Kelso, Dynamic pattern generation in behavioral and neural systems, *Science*, 239:1513-1520, 1988.

[51] G. Schöner, P.G. Zanone, and J A.S. Kelso, Learning as change of coordination dynamics: Theory and experiment, *Journal of Motor Behavior*, 24:29-48, 1992.

[52] E.R. Schulze, S. Bohrer, M. Dose, and S. Fuhrmann, An active vision system for navigation in a natural environment, In *Proc. IJCNN, San Diego*, pages II:729-234, 1990.

[53] Tim Smithers, What the dynamics of adaptive behaviour and cognition might look like in agent-environment systems, In *Presented at the workshop: On the Role of Dynamics in Representation in Adaptive Behaviour and Cognition, 9/10 December 94, San Sebastian, Spain*, 1994.

[54] L. Steels, Exploiting analogical representations, *Robotics and Autonomous Systems*, 6:71 - 88, 1990.

[55] L. Steels, Mathematical analysis of behavior systems, In P. Gaussier and J-D Nicoud, eds., *Proc. From Perception to Action Conf., Sep. 7-9, 1994, Lausanne, Switzerland*, pp. 88-95, Los Alamitos, CA, 1994. IEEE Computer Society Press.

[56] A. Steinhage and G. Schöner, in preparation.

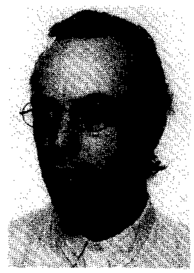
[57] K. Storjohann, T. Zielke, H.A. Mallot, and W von Seelen, Visual obstacle detection for automatically guided vehicles, In *IEEE Proceedings of the 1990 IEEE International Conference on robotics and automation*. IEEE Computer Society Press, 1990.

[58] E. Thelen and L.B. Smith, *A Dynamic Systems Approach to the Development of Cognition and Action*, The MIT Press, A Bradford Book, Cambridge, Massachusetts, 1994.

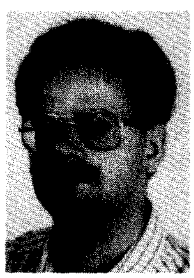
[59] R.B. Tilove, Local obstacle avoidance for mobile robots based on the method of artificial potentials, In *Proceedings of the IEEE Conf. Robotics and Automation, Cincinnati, Ohio, May 1990*, pages 13-18 and 566-571, 1990.

[60] M.T. Turvey, Preliminaries to a theory of action with reference to vision, In R.E. Shaw and J. Bransford, editors, *Perceiving, acting and knowing*, pages 211-265, Hillsdale, New Jersey, 1977. Erlbaum.

[61] S. Zrehen and P. Gaussier, Why topological maps are useful for learning in an autonomous agent, In P. Gaussier and J-D Nicoud, editors, *Proceedings From Perception to Action Conference, September 7-9, 1994, Lausanne, Switzerland*, pages 230-241, Los Alamitos, California, 1994. IEEE Computer Society Press.



Gregor Schöner (1958) obtained a PhD in theoretical physics from the University of Stuttgart in Germany in 1985. For the next four years he applied his background in stochastic dynamical systems to do theory on the coordination of biological movement at the Center for Complex Systems in Boca Raton, Florida. From 1989 to 1994 he led a group at the Institut für Neuroinformatik in Bochum, Germany, in which ideas from dynamical system theory were developed for such diverse fields as perception, movement, cortical neurophysiology, and autonomous robotics. He is now at the Laboratory for Cognitive Neuroscience in Marseille, France, where he pursues an interdisciplinary research agenda in these same fields centered around the theoretical concept of neural field dynamics.



Michael Dose (1961) was born in Düsseldorf, Germany. He studied computer science and physics at the University of Dortmund, Germany, where he has received his diploma degree in computer science in 1989. Since then he has been working as a research assistant at the Institut für Neuroinformatik in Bochum, Germany, receiving his doctoral degree in computer science in 1994. His doctoral work concerned autonomous mobile robot path planning, architectures for autonomous systems and theory of biological movement. Since the beginning of 1995 he is working in the domain of computer vision.

Christoph Engels (born in 1965 in Wuppertal, Germany) studied computer science and electrical engineering at the Universität of Dortmund, Germany, where he received his diploma in computer science in 1991 with a thesis about visual object recognition. Then he moved to the Institut für Neuroinformatik at Ruhr-Universität-Bochum in Germany where he has worked as a research assistant with a doctoral thesis about dynamic fields for autonomous robots. Now he is an employee at the Zentrum für Neuroinformatik GmbH in Bochum in the field of economic forecasting with neural networks.